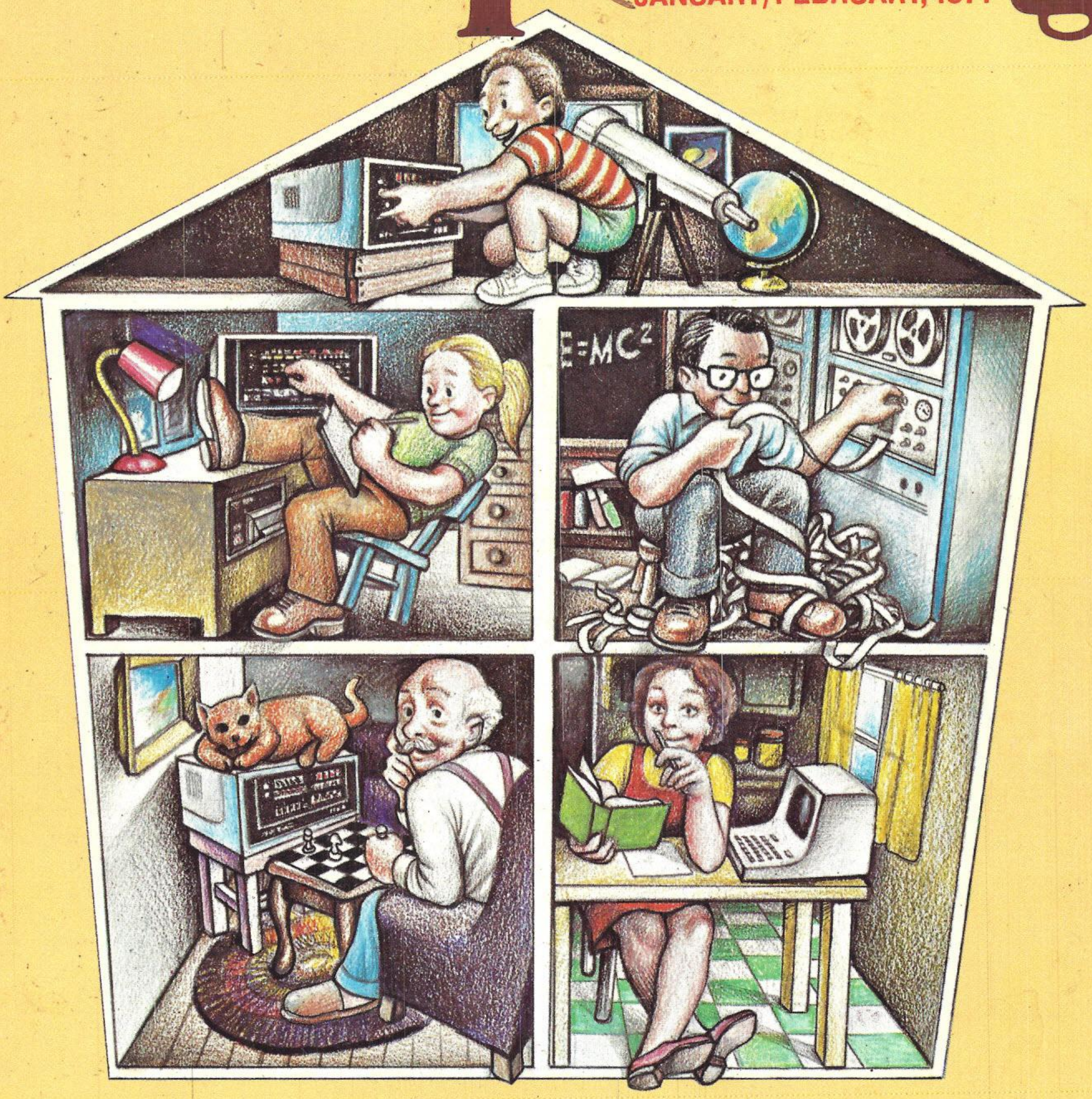
$2.00

Personal Computing

JANUARY/FEBRUARY, 1977



Computers are for everyone!

SPAGHETTI BASIC ▪ HARD TALK ON HARDWARE
LEMONADE COMPUTER SERVICE COMPANY
FREE COMPUTER ART ▪ COMPUTER CRIME

PREMIER ISSUE

Try pricing out the cost of lost data.

Then you'll know why over half the certified digital cassettes in use throughout the world today are made by Information Terminals.

A dropped bit could mean only a misspelled name. It could also mean a check a hundred or a thousand times too big, a carload shipment sent to the wrong destination, or a fleeting moment in time lost forever!

100% Certified. You can be sure with ITC.

To help protect you, your company and your projects against such losses, ITC tests its computer grade information storage media 100% *after assembly.*

This means *every* ITC computer grade digital cassette, floppy disk, FLIPPY™ disk, ¼-inch Data Cartridge and Mini-Cassette is checked for perfect mechanical and recording operation before it leaves our hands. 100% Certified. You're sure with ITC.

A medium for every message.

ITC 100% CERTIFIED DIGITAL CASSETTES. Five Series, to fit just about every system and environmental requirement, from computer room or business office to desert, jungle or frigid mountain top.

ITC 100% CERTIFIED FLOPPY AND FLIPPY™ DISKS. One-sided floppies and two-sided FLIPPIES™. No matter what or who's diskettes you've been using, there's a 100% Certified ITC disk you can depend on.

ITC 100% CERTIFIED ¼-INCH DATA CARTRIDGES. ITC's new ¼" Data Cartridges, with exclusive new features for new levels of performance and reliability.

ITC 100% CERTIFIED MINI-CASSETTES. A unique new approach to data storage in hand-held and desktop digital devices. ITC's MI-50 Mini-Cassette packs 50 feet of data capture in a package smaller than a box of pocket matches.

ITC MAGNETIC CARDS FOR WORD PROCESSING SYSTEMS. High quality products for the Office,

from the leading producer of 100% Certified computer grade magnetic media. Like all ITC products, our Mag Cards feature consistently high performance from one shipment to the next.

Write for Information on ITC computer grade and word processing media, and our cassette and diskette testing and alignment equipment. Or call your nearest ITC distributor or representative.

Information Terminals
323 Soquel Way
Sunnyvale, California 94086
Telephone (408) 245-4400

In Europe:
Information Terminals SA
Case Postale 296
1215 Geneva
Switzerland

CIRCLE 4

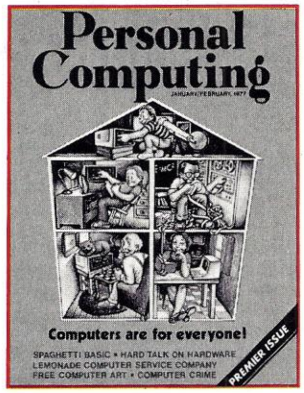
Personal Computing

JANUARY/FEBRUARY 1977 VOL.1 NO.1

FEATURES

DEPARTMENTS

COVER



Kim Behm's picture of the modern home: a personal computer in every room.

Subscription rates (annual): U.S., $8; Mexico and Canada, $14 surface mail and $18 airmail; All other countries, $26 airmail. Single copies and back issues (as available): $2 in U.S.; $4 in Canada and Mexico; $5 in all other countries. Send subscription orders and changes of address to: Circulation, Personal Computing, 167 Corey Road, Brookline, MA 02146.

Personal Computing is published bimonthly by Benwill Publishing Corp., 167 Corey Road, Brookline, MA 02146. Harold G. Buchbinder, Chairman of the Board; George Palken, President; Esther Shershow, Treasurer. Editorial office: 401 Louisiana S.E. No. G, Albuquerque, NM 87108; telephone (505) 266-1173. Publication office: 167 Corey Road, Brookline, MA 02146; telephone (617) 232-5470. Application to mail at second-class postage rates is pending at Brookline, MA 02146.

STAFF

Imagine a microcomputer

Imagine a microcomputer with all the design savvy, ruggedness, and sophistication of the best minicomputers.

Imagine a microcomputer supported by dozens of interface, memory, and processor option boards. One that can be interfaced to an indefinite number of peripheral devices including dual floppy discs, CRT's, line printers, cassette recorders, video displays, paper tape readers, teleprinters, plotters, and custom devices.

Imagine a microcomputer supported by extensive software including Extended BASIC, Disk BASIC, DOS and a complete library of business, developmental, and industrial programs.

Imagine a microcomputer that will do everything a mini will do, only at a fraction of the cost.

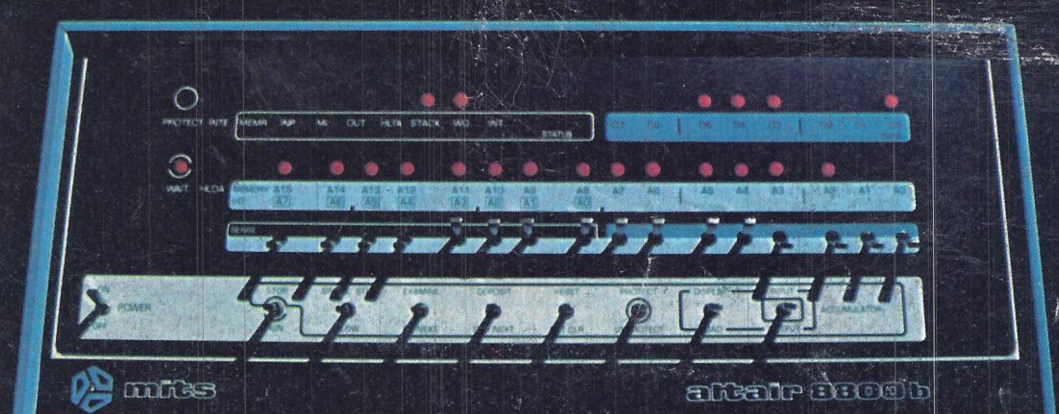
You are imagining the Altair™ 8800b. The Altair 8800b is here today, and it may very well be the mainframe of the 70's.

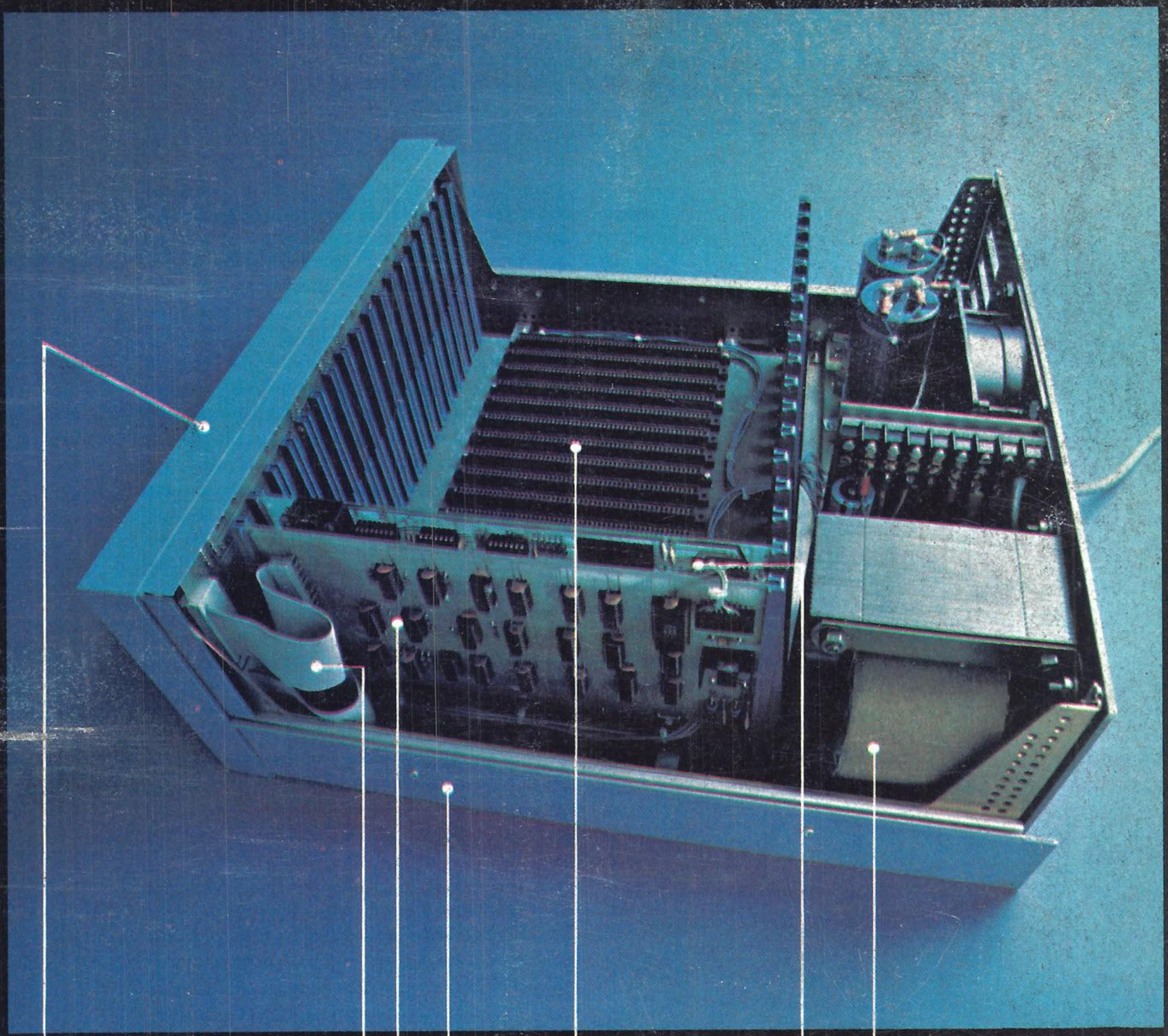
The Altair 8800b is a second generation design of the most popular microcomputer in the field, the Altair 8800. Built around the 8800A microprocessor, the Altair 8800b is an open ended machine that is compatible with all Altair 8800 hardware and software. It can be configured to match most any system need.

MITS' plug-in compatible boards for the Altair 8800b now include: 4K static memory, 4K dynamic memory, 16K static memory, multi-port serial interface, multi-port parallel interface, audio cassette record interface, vectored interrupt, real time clock, PROM board, multiplexer, A/D convertor, extender card, disc controller, and line printer interface.

MITS' peripherals for the Altair 8800b include the Altair Floppy Disc, Altair Line Printer, teletypewriters, and the soon-to-be-announced Altair CRT terminal.

Introductory prices for the Altair 8800b are $840 for a kit with complete assembly instructions, and $1100 for an assembled unit. Complete documentation, membership into the Altair Users Club, subscription to "Computer Notes," access to the Altair Software Library, and a copy of Charles J. Sippl's Microcomputer Dictionary are included.





Redesigned front panel. Totally synchronous logic design. Same switch and LED arrangement as original Altair 8800. New back-lit Duralith (laminated plastic and mylar, bonded to aluminum) dress panel with multi-color graphics. New longer, flat toggle switches. Five new functions stored on front panel PROM including: DISPLAY ACCUMULATOR (displays contents of accumulator), LOAD ACCUMULATOR (loads contents of the 8 data switches (A7-AO) into accumulator), OUTPUT ACCUMULATOR (Outputs contents of accumulator to I/O device addressed by the upper 8 address switches), INPUT ACCUMULATOR (inputs to the accumulator from the I/O device), and SLOW (causes program execution at a rate of about 5 cycles per second – for program debugging).

Full 18 slot motherboard.

Rugged, commercial grade Optima cabinet.

New front panel interface board buffers all lines to and from 8800b bus.

Two, 34 conductor ribbon cable assemblies. Connects front panel board to front panel interface board. Eliminates need for complicated front panel/bus wiring.

New, heavy duty power supply: +8 volts at 18 amps, +18 volts at 2 amps, -18 volts at 2 amps. 110 volt or 220 volt operation (50/60 Hz). Primary tapped for either high or low line operation.

New CPU board with 8080A microprocessor and Intel 8224 clock generator and 8216 bus drivers. Clock pulse widths and phasing as well as frequency are crystal controlled. Compatible with all current Altair 8800 software and hardware.

altair 8800-b

CIRCLE 12

mits
2450 Alamo SE/Albuquerque, NM 87106/505-243-7821

MEMO FROM THE PUBLISHER

Publisher David Bunnell dates his introduction to computers to November 1973, when he walked into his boss's office at Mits, a small electronics firm in Albuquerque, N.M. At the time, Bunnell was employed as a technical writer. The boss, H. Edward Roberts, had asked him to come up with a list of possible names for Mits' newest product. It was a computer that you put together from a kit. Having spent half the night searching through Roget's Thesaurus, familiarizing himself again with the Greek gods he had studied in school, Bunnell felt that somewhere on his list was a winner.

At that time, most of the 15 or 20 employees at Mits, including Bunnell, thought of computers as three or four standup freezers tied together with bailing wire. How Mits was going to produce such a thing in kit form with an "easy to assemble" instruction manual was beyond them.

"God knows what people will do with this thing," was one of Bunnell's earlier thoughts.

"Oh that," Roberts said when Bunnell handed him the list of names. "That's already taken care of. I just talked to Les Solomon at *Popular Electronics.* His daughter came up with the name 'Altair'. She was watching Star Trek and the Enterprize was going to the Altair star. It's a first mannitude star and maybe an ancient god. I like it. What do you think?"

Bunnell liked Altair just fine. It wasn't on his list, but he was off the hook. If the thing turned out to be a flop, at least he couldn't be blamed for naming it. That December, Bunnell was promoted to Vice President of Advertising at Mits. The Altair 8800 computer, introduced in January of 1975 on the cover of *Popular Electronics,* was an instant success.

As PERSONAL COMPUTING's publisher, Bunnell no longer has any ties with Mits. He is at work though on an insider's account of Mits. This account, entitled "The Age of Altair", will be published in book form. Look for excerpts in PERSONAL COMPUTING.

A computer lies in your future

If you don't have a computer by now, chances are there's one in your immediate future. By the early 1980's, many experts predict that every middle-class home will have a home computer system. This system will be used to control your sprinkler system, do family accounting, act as a teaching tool for you and your children, monitor and control energy consumption and provide you with immediate access to whole libraries of information. It will also challenge you to play an infinite number of mind-boggling computer games.

Today, over 15,000 personal computers are up and running. Thousands more are in the works. If you purchased a computer two years ago, you probably had to buy through the mail. If you purchased it six months ago, you probably got it from one of the dozens of retail computer stores that have opened up throughout the country. The growth of this market is truly phenomenal. New microcomputer products are introduced weekly. The home computer is having an impact on foreign markets, particularly Canada and Europe. A recent Personal Computing convention drew over 5,000 computer enthusiasts.

The key to continued success in personal computing can be stated in one word: *simplicity.* Price is no longer the issue. You can buy a good personal computing system for about the price of a good stereo system. Before you buy a computer though, you will want to feel confident that you can understand your system, and most importantly, make use of it.

We created PERSONAL COMPUTING to help fill the gap between computer technologists and the general public. We strive to reassure you about the ease with which you can master your own, personal system. It is important to us that your personal computer be more than a prepackaged black box. Personal computing should be individual computing.

To use a personal computer, you need not be a programmer or an electronic technician. However, you should be able to use your system effectively, modifying it to suit your own, personal needs. You need to know how to program in BASIC (or whatever "Better Than BASIC" language that comes along). You may not be writing long, complex programs, but how much better it will be if you can add subroutines to your prepackaged family accounting program.

Have you ever considered the multitude of advantages a good personal computer system could be to a farmer? Or to a housewife? A student? Small businessman? Or to yourself, for whatever it is that you do.

PERSONAL COMPUTING established the Lemonade Computer Service Company to point out the many advantages computers offer to various groups of people. If you're an enterprising person, we even hope to show you how to turn a few bucks with your own system.

While it is our policy to aim editorial content at those of you outside the professional computing business, we hope that those of you who are computer professionals will find our magazine entertaining. We hope you'll be able to glean

some good ideas, and that you'll find each issue of our magazine a handy resource of what's happening in the personal computing market.

Today's personal market

Today, at least five computer mainframes on the market have the same bus structure as the original Altair 8800. Over 100 plug-in circuit boards are available to support any one of these mainframes. These boards reflect a wide range of engineering talent, research dollars, etc., of a dozen different companies. They include 4K, 8K and 16K memory boards; serial and parallel interface boards; A/D converters; PROM memory boards; video interface boards; floppy disk controllers; line printer controllers; multiplexers; and many more.

It is the collective support of the Altair bus that diminishes the spectre of a large company coming along and wiping everyone else out. True, the personal computing market is becoming more competitive everyday (see our Atlantic City convention report). Several large companies are now taking a serious look at the market. However, some of the small companies have a good chance to survive and prosper.

What excites me most about the situation in today's personal computing market is the emergence of a good variety of low-cost hardware. PERSONAL COMPUTING is establishing an independent consumer evaluation team to look at this hardware, report on it and make recommendations to our readers.

You can count on PERSONAL COMPUTING to be fair and honest in its reporting. It is our strict policy to keep editorial and advertising completely separate operations. You won't find product reviews written by employees of the companies that produce the product. And you won't find that we've printed such an article as a ploy for attracting a company's advertising.

Aside from hardware, there are signs that software and applications are beginning to assume a greater priority. Many dealers are promoting small business application packages.

This market is ready to really explode. 1976 has been a good year for personal computing. 1977 will dwarf it by comparison.

The end of the kit?

Consider this: for every hobbyist out there willing to build a computer kit there must be a 100 potential hobbyists who simply don't have the time or don't feel confident with a soldering iron. These people want to play Star Trek, write their own check balancing programs and explore the possibilities of using computers in their businesses, but they are not about to build a kit.

While in the short term, computer kits are a way of rapidly expanding the market, in the long run they may inhibit market growth. How much further along would personal computing be today if half the energy spent on kit building instead went into applications?

But kits are cheaper, you say. No, the price difference between kits and assembled units is artificial.

You see, a small company that lacks a good production line can deliver hundreds of computer kits a month while it may be able to assemble only a handful. However, while these companies save labor costs, they incur other costs that begin to balance out the cost of an assembly line. These costs are in technical writing, communications with customers and in the repair department.

The repair department is the most telling area. For every assembled computer product returned under warranty, there are upwards to 100 kit computer products. True, most of these repairs are for solder bridges and the like, but they add up to a lot of bucks.

Personal computer manufacturers have seen a growing percentage of their sales go to the oem and small business markets (despite the artificial price difference between kits and assembled units). Thus, they have been forced to build up their assembled production capacity. This has made them realize that: (a) there is more profit in assembled units and (b) customer relations are greatly improved when you're shipping *tested* units.

Look for big reductions in the prices of assembled computer gear in the coming months.

Is this the October issue?

If you read our advertisements, you are probably aware that this was to have been the October/November issue of PERSONAL COMPUTING. Those of you who are subscribers to PERSONAL COMPUTING are probably wondering if you'll get the premier issue plus a year's subscription as promised. (The answer is "yes". Your subscription has automatically been extended to include the first issue of 1978.)

Let me explain.

PERSONAL COMPUTING is a consumer magazine that depends on newsstand sales for a large part of its circulation. While this issue was indeed printed in October, to have put "October/November" on the front cover would have crippled newsstand sales. That is because by October most of the magazines on the newsstand are November, December and even January issues. So who's going to buy an October/November issue of PERSONAL COMPUTING when they can buy a January issue of whatever?

We could have called this issue the "November/December" issue, but that would be cutting it a bit too close. By the second week of December, magazine vendors would have been discarding any remaining issues of PERSONAL COMPUTING. Since this is the premier issue, we want it to be on the stands long enough for people to see, think about and of course buy.

Actually, "December/January" would have been ideal. Ideal for everyone except the bookkeeper who would have been faced with the accounting problem of a product that is halfway in one fiscal year and halfway in another.

So welcome to the "January/February" issue!

Buffalo games by mail

The Flying Buffalo has rounded up a thousand or more people around the world to play games created by Richard Loomis, Box 1467, Scottsdale, Ariz. 85252.

Loomis decided several years ago that the complex games he liked to play (Monopoly in many variations and the numerous battle games offered by various companies) could be made much more complex and satisfying with a computer keeping track of everything. He therefore devised elaborate games involving calculation superbly suitable for a computer.

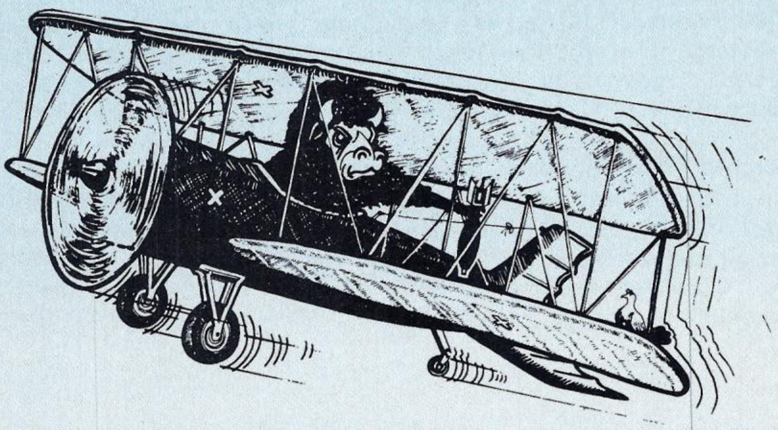
Released from the service, Loomis put himself into deep hock to buy a computer and offer participation in the games commercially. He's had enough takers to become self-supporting, that is, Loomis-supporting.

In the games played by mail, each participant pays a modest set-up fee, gets rules to study and then mails in his first move by a deadline. After running the moves through a computer program, Loomis sends players their results and detailed information about their standings. The players respond . . . Loomis responds . . . and so on until the game is completed. Typically, each player makes two moves a month at fees ranging from thirty cents to a dollar a turn. A major game can run for 40 weeks.

What's so special about a game that takes two weeks a move? The scope. Players operate in a vast imaginary universe, make decisions that affect the destinies of huge populations and warp history both generally and in detail for millenia to come. The computer allows practical manipulation of richly embroidered fantasies that occupy the games players intensely.

In StarWeb, for example, a player may choose any role among six possibilities. He may be Empire Builder, Merchant, Pirate, Artifact Collector, the Berserker or the Apostle. The characteristics of the Berserker are remarkably similar to those of the devices in Fred Saberhagen's famous science fiction stories, although Loomis claims he chose the word from a dictionary. The Apostle wins by converting everybody to his religion, loses points by killing anyone and gains points when his followers are killed (become martyrs).

The object of the StarWeb game is to win hegemony over the greatest number of the 255 star systems in the game. The player, exploring and mapping the universe as he goes, meets friends and foes. He's never certain how many are playing the game and never fully controls his environment. If he's a dedicated game player, the computer opens worlds of exotic entertainment.



Flying Buffalo, Inc. offers a variety of games-by-mail, table games, T-shirts and newsletters. If you have worked out a grand fantasy of your own, complete with debugged programs and rich descriptive literature, maybe the Buffalo's marketing experience can help you.

Computer pictures - 3¢ each

Artist and Computer
Ruth Leavitt, ed.; Creative Computing or Harmony Books Div. Crown Publishers, New York, N.Y.; $4.95

Ruth Leavitt's compilation of 160 pieces of computer art is the only up-to-date computer artbook. In it, 35 newcomers and old masters such as Ken Knowlton, John Whitney and Charles Csuri display their works and comment on them.

Much has happened in computer art since the days described in Jasia Reichardt's **Cybernetic Serendipity** and Herbert Franke's **Computer Graphics — Computer Art**. The cold mathematical beauty of delicately drawn geometric shapes and processed pictures has given way to broad swaths of color and free-flowing human-like shapes. As demonstrated in this book, each of the 35 artists has developed an individual style, and the most versatile artists juggle several styles at once.

In her preface, editor Leavitt emphasizes the book's variety. "The computer can function for the artist at many different levels. The artist has only to choose what role he/she wishes the computer to play. Apart from produc-

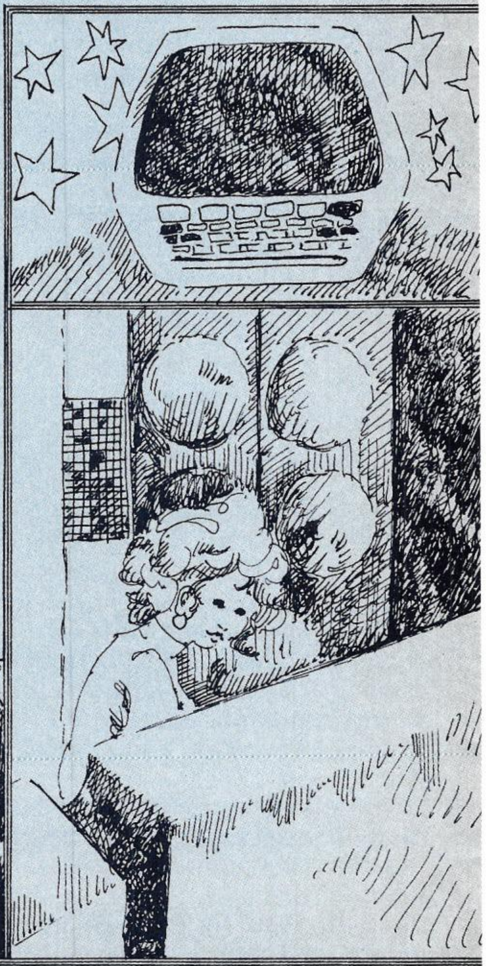
random access

ing finished pieces of artwork, as William Kolomyjec does, one may simply allow the computer to function as an idea machine. This is evidenced in several articles. Karen Huff, for example, describes how the computer is used to visualize fabric before it is actually woven . . . Robert Mallary creates sculpture with his program Tran 2 . . . A 3-dimensional program called Shape 3/D is used as a tool in research of aesthetics and art theory . . . Edward Ihnatowicz is deeply interested in artificial intelligence and uses this approach in creating cybernetic sculpture. Aaron Marcus' work shows his interest in concrete poetry. He creates picture environments, and his poem drawings give new meaning and depth to words. Duane Palyka uses a color television monitor attached to a computer system and paints pictures in an unprecedented manner. Colette and Charles Bangerts' drawings appear hand-made. Aldo Giorgini's moire patterns have the look of optical art. Each takes advantage of different features offered by the computer."

Responding to Leavitt's elaborate questionnaire, each artist describes his feelings about working with the computer. She says, "It is important for artists to be able to discuss their work in their own words. Therefore, the papers in this book are presented with a minimum of editing." Frankly, I wish she had done more editing, because reading 35 responses to a questionnaire gets a bit tiring. The 120 pages are organized in 35 sections — one per artist — in a mysterious non-alphabetical order. She should have set up some themes to pull the book together, beyond the statements in her excellent brief preface. Her book displays just artists' output and philosophies; it should also have described their specific hardware and programming techniques. The decision to shoot only 12 of the plates in color is disappointing.

Leavitt compiled the book at the bidding of Dave Ahl, editor of **Creative Computing** magazine. He distributed the book to the magazine's subscribers in place of two summer 1976 issues. You can order it through the magazine or buy it directly from Crown Publishers.

—RW



Someday a star will rise

You'd expect show-business celebrities with enough cash in their jeans to buy whatever they like to move into personal computing with great enthusiasm. After all, television and motion pictures are highly technical media dependent upon a large corps of competent technical people for day-to-day successful operation.

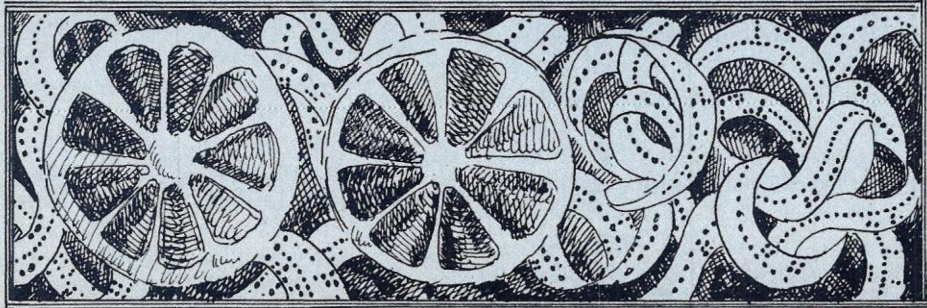
Anybody who has worked in film or television knows that many of the celebrities are extremely intelligent, hard-working people. Success on the silver screen is not always a matter of dumb luck.

So where are the celebrities?

A modest search by us has turned up a few writers and producers with systems of their own, but no household names, no stars who admit to personal hobby computer interest. Maybe they buy computers under assumed names or through their agents. But no dealers or manufacturers consulted were aware of famous folk behind the buyers of record.

Odd.

Feature stories on celebrity-owned and -operated computer systems might be entertaining and informative occasionally . . . if there were any.

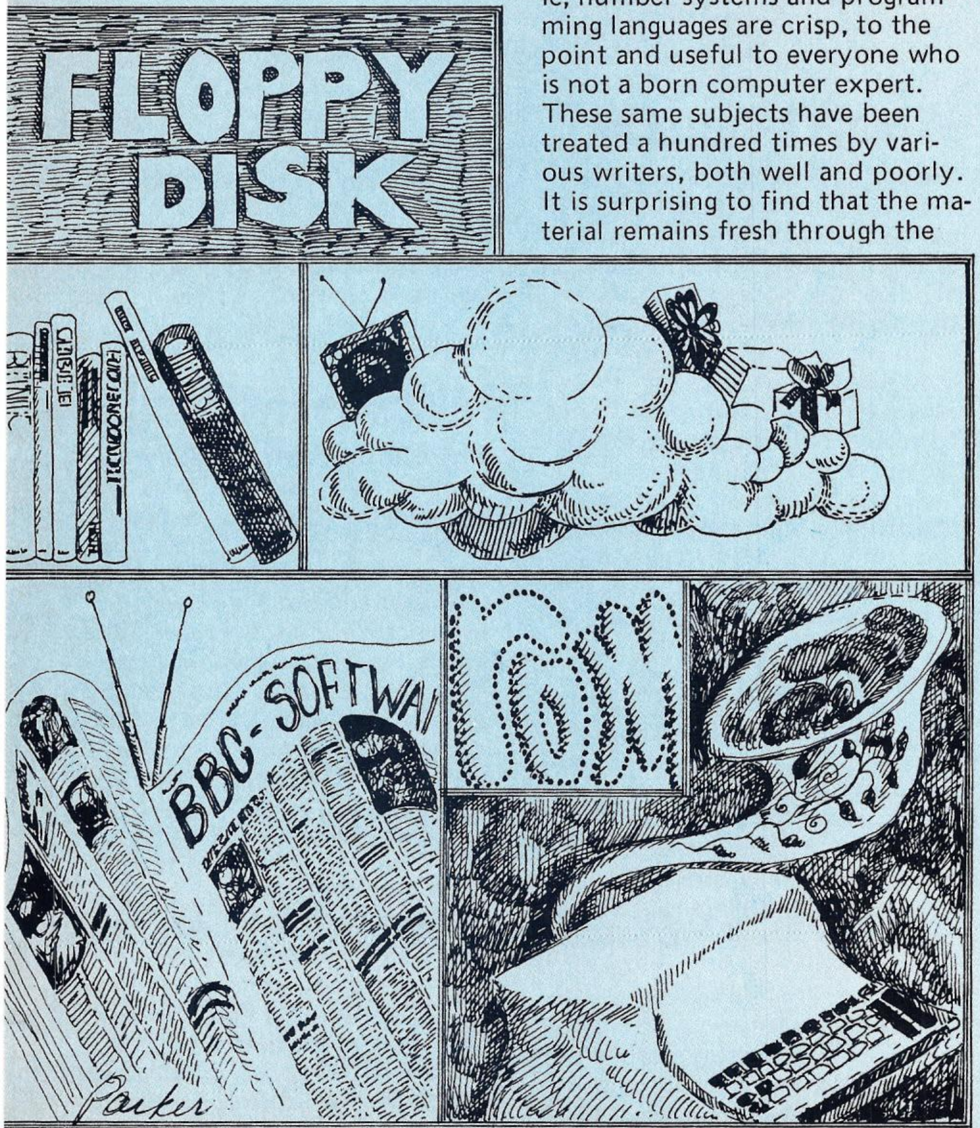


Hardware primer

Hobby Computers Are Here

Wayne Green, ed.; 73, Inc., Peterborough, N.H., $4.95

Wayne Green has done a helpful thing. He has rummaged through back issues of his **73** magazine, selected 21 basic articles on logic, hardware and software, mingled them with 21 of his editorials and offered them for sale in the form of this paperbound book, which is magazine size.



The resulting volume is a useful and entertaining primer for hardware-oriented hobbyists. The elektroniker will be cheered to find schematics by the score in articles treating practical and instructive projects in ASCII to Baudot conversion, application of keyboards to display systems and the ins and outs of dealing with Morse code. The non-elektroniker may not be aided much by photographs of traces on oscilloscopes, but he does get an invigorating dip in the realities of hardware design.

The tutorial pieces on TTL logic, number systems and programming languages are crisp, to the point and useful to everyone who is not a born computer expert. These same subjects have been treated a hundred times by various writers, both well and poorly. It is surprising to find that the material remains fresh through the years, because each writer's insights and explanations have unique character that illuminates the subject. This is "just another collection among many," but it's a good one. Worth having.

And the editorials? Green is a fearless prophet predicting, even demanding, the development of new systems for new applications. He's indignant about the slow development of national facsimile transmission systems at reasonable cost and imagines that microcomputer technology will change all that. He's big on computers in HAM systems, small business applications and fresh orange juice. Green deals in informed conjectures on everything that seems to touch the non-professional computer people. In a fit of enthusiasm, he points to a $10 billion to $50 billion a year market that will rise from the modest, if remarkable beginning of the amateur computing field. Bold talk.

Bristling with numbers, educated guesses, reminiscence and personal feuding, Green's editorials would be tiresome chatter, hardly worth reading, let alone preserving, if they weren't well-written. They are well-written, straightforward, disciplined in style. Unlike many a technologist who has lately figured out how to operate a typewriter, Green is literate and sensitive to his reader's interests.

The book is nicely laid-out, too, attractive, easy to read, with photographs carefully printed and diagrams neatly rendered. That's a novelty in the hobbyist field. The casual reader can figure out what he's looking at without extraordinary effort.

Wayne Green's hobby computer book has immediate practical value and will have historical interest value as the years pass and we all learn how the fearless prophecies fared. **—NBW III**

random access

Boob tube software

You found the software somewhere, and now you've got to load it into your machine. Will Overington of Evesham, Worcestershire, England suggests televised software to ease your load. Here's his concept.

"Imagine that your microprocessor-based home computer was linked to a 'read-only disc' from which software packages could be obtained as desired. Imagine further that this 'disc' did not exist as a piece of hardware in your installation but was in fact sited at a uhf television transmitter. With the output of the disc continually transmitted, your computer would merely look at a continually arriving bit stream and, when the package it required started to arrive, begin to store the subsequent microcode in random-access memory, ready to be obeyed when required. I call this concept **Telesoftware.**

"British television companies are running a pilot trial of a novel type of broadcasting known as teletext. The idea is to produce a ' television newspaper ' service. Digitally coded alphanumeric and graphics data are transmitted in some of the lines of the otherwise unused field blanking interval of a conventional uhf television transmission. In order to use teletext signals a special module needs to be added to the receiver. Pages of the newspaper are transmitted sequentially. One 40-character line and its address fit into a field blanking line. Each page has 24 lines.

"When the desired page, indicated by using a hand-held keyboard, comes round, it is captured line by line and stored in a random-access memory. A display controller with a character-generation, read-only memory produces a display in white and six colors on the screen. Maximum access time is about 20 to 30 seconds. Pages may be updated and the page selection varied by the broadcasting company.

"Telesoftware essentially proposes that the random-access memory of a teletext receiver module is made a subset of the memory of a microprocessor system. Bytes of software would then be obtained by passing pairs of teletext characters through a hardwired Hamming corrector. Results of computations would be displayed on the television screen, often by inserting the user's results into blanks in a conventional teletext page. Telesoftware could be implemented using standard teletext transmission hardware. No telephone wires are needed and since the system is ' one-way ' there is no limit on the number of terminals which could be used simultaneously.

"I have not patented telesoftware as I do not wish to impose restrictions on its use. If implemented, it could provide software for educational purposes and TV games."

What'd he say?

There's a lot of talk about inexpensive speech synthesis systems compatible with now-standard microcomputers that are in the hands of the hobbyists. A couple of systems are on — or almost on — the market already. Type your messages to the computer, the idea goes, and it will speak back to you aloud through your hi-fi system or your old radio.

Notice that most of the talk is **about** these inexpensive systems, not **from** them. Experimenters report that the synthesized speech is interesting, all right, but awfully difficult to understand. Before you plunge with your money, these pioneers say, it's a good plan to listen to the system output and determine by test whether the system is satisfactory or not. A taped demonstration is not always a good guide, since even expensive systems seem to call for tender-loving-care to make them perform well. You'd like to know ahead of time just how much tweaking-up of knobs and pots is necessary for intelligibility.

The bugs are being worked out, one by one, and the speech synthesis systems are real. Rumor even has it that a genuinely inexpensive, flexible speech **recognition** system has been developed so the computer can take spoken instructions as well as give backtalk at an amateur's price.

Missing Misses

Why is it that observers have so far uniformly reported to us that women are not widely in evidence in this field? One publisher of a newsletter reports he's had but a single letter from a woman in five years. A store operator was struck by the observation that few of his customers are women. A manufacturer searched records in vain to find more than a tiny percentage of women among the customers.

Surely, there is no artificial barrier to women who want to be hobbyists. Surely, women are in the forefront of the programming field with professionals like Jan Stone, Leigh Hendricks and Brenda Bittner doing original, effective work even more than a decade ago.

Maybe the reports are mistaken. Our survey was casual. Does anyone have statistics to offer?

It has always been respectable for young people to undertake some commercial enterprise that demands work, a strong sense of responsibility and an element of risk. Newspaper routes have been traditional launching pads for businessmen as well as the lemonade stand, a legendary symbol of youthful entrepreneurship. (Old joke about this: Man drinking lemonade at one of two side-by-side stands says to the kid behind the counter: "Why do you sell your lemonade for only three cents a glass when the vendor next to you asks ten cents a glass? Surely the cutthroat competition is ruinous to both of you." "Oh, well," says the three-cent kid, "the cat didn't fall in *his* lemonade bowl.") Lemonade stands are outstandingly respectable in our society.

That aside for the moment, consider this question:

"What would young Stanislaus do with a computer if he had one?"

Young Stanislaus, a junior high school student, has just suggested investing his ten-year accumulation of banked birthday-present money in a personal computer system. In a whining tone, Stan's aunt has put the question above to all the household's adults. She thinks of a screwdriver as complex technology.

Fun & Games

Unknown to her, Stan plans to play a lot of games with his system, learning its capabilities and characteristics by making the most of the flood of materials that game-loving programmers pour forth constantly. However, the lad knows full well he must persuade Aunt Yadwiga that he is seriously preparing himself for the future, earning money or improving the world's moral outlook, just to keep her from nagging him and his parents to the point of distraction.

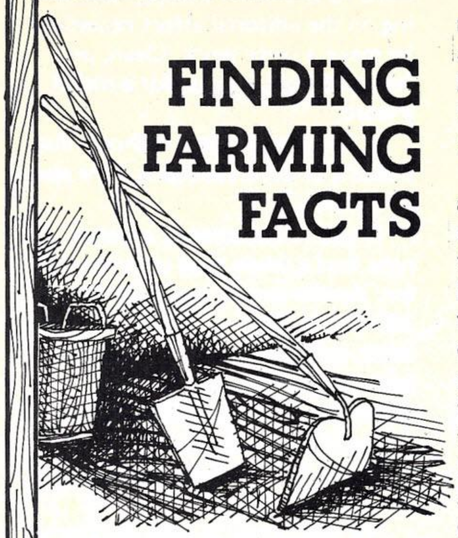
He casts about briefly for some respectable excuse for buying his computer system with his own money and sensibly establishes the LEMONADE COMPUTER SERVICE COMPANY, a noble and dignified sole proprietorship.

Reports on the activity of the company may serve as guides to other desperate beginners who must explain sensibly the irrational urge they feel to have personal computers. (Ignorant and uncertain adults who make use of these notions can explain their interest away as concern for youth.)

Stan's purpose is to perform useful services for the neighbors at modest fees, using his personal computer. He's game for anything.

Let us consider a LEMONADE venture into gardening.

FINDING FARMING FACTS



The entrepreneur who wants to operate a Lemonade Computer Service Company Gardening Guide will require far more data than offered in this brief, conceptual article. Any public library can provide numerous books to help while away the long winter nights in preparation for spring.

In addition, Superintendent of Documents (U.S. Government Printing Office, Washington, D.C. 20402) can provide a catalog of excellent pamphlets the government has prepared at staggering expense to inform interested parties on every imaginable aspect of gardening. Some of the items are available free. Others may be purchased for a pittance. Invaluable material. Your County Agricultural Extension Agent (see your phone book) may also be able to steer you to valuable information of importance in your locality.

The people who sell seeds, bulbs and seedling plants can also give you magnificent, tantalizing catalogs of their wares, loaded with important information about specific vegetables and flowers. A courteous request for assistance and guidance may produce more information from the commercial firms than they commonly distribute, especially if they feel that their interests are being served. PERSONAL COMPUTING has made no effort to communicate with these companies, so you're on your own. Here's a sampling of well-known firms:

W. Atlee Burpee Co.
50266 Burpee Building
Warminster, Pa. 18974

Gurney Seed and Nursery Co.
Yankton, S.D. 57078

Northrup King Seeds
Minneapolis, Minn. 55413

Asgrow Mandeville Co.
Cambridge, N.Y. 12816

Henry Field Seed and Nursery Co.
Shenandoah, Iowa 51601

Ferry-Morse Seed Co.
Fulton, Ky.
or Mountain View, Calif. 94040

Green Thumb Advisory

The number of private flower and vegetable gardens is reportedly on the increase, what with rising food prices and a national mania for getting back to dirt. Not all gardeners have green thumbs or even common sense, and they may welcome a qualified personal advisory service tailored to their tastes, needs and resources. So that all the labor with a shovel is productive, they'd be glad to pay a bright kid $10 or $15 a season for detailed advice on planning and management of their back forty.

During the long, chilly winter, young Stanislaus not only does his homework and plays Star Trek on the computer but collects immense numbers of useful pamphlets and papers from the U.S. Department of Agriculture, seed companies and university experts. The materials provide masses of data about standard garden vegetables and flowers.

The data tell Stan how well carrots, for example, grow in what kinds of soil, how much moisture they require and at what rate, the amount of sunlight and shade they tolerate, which fertilizers suit them, what diseases and insect plagues befall them, how fast they grow, what their protein and vitamin content is, and how they may be harvested and preserved. The booklets even suggest how many carrots are required to feed one person his fill of orange roots for a given period of time.

What About Okra?

The literature answers baffling but important questions like: how many button squash can be obtained from a single hill in a season? How many squash hills are required to satisfy a family of four that likes squash? What about okra? How many peas are enough?

. . . and so on. The picture is clear by now. Stan carefully transfers all the available data about conventional crops to his computer system. From a client's garden size, taste, needs and budget, Stanislaus can use a program he has written to have the computer plan a satisfactory garden. In no time flat, Stan can offer half a dozen alternative plans to the client. An elegant start.

Come planting time (determined by the computer system from input data on temperature, humidity and the date), Stan makes the rounds of the plots he is supervising and takes soil samples, which he runs down to a local laboratory for analysis. The lab operator cuts his rate slightly for the quantity of samples Stan brings in, so the lad turns an extra dollar or two on the transaction. Good business.

Since the computer system has been told all it needs to know about moisture, temperature, fertilizers and insecticides, it can print out daily specifications for action required in each garden for each crop. That is, if Stan tells it each day about conditions. If he is as resourceful and energetic as the average successful paper boy, he visits each plot daily, makes notes on a form he has prepared for the purpose and passes out computer-printed instructions to the gardeners.

Tolerant Computer

Note well that Stan handles information, not shovels. No backbreaking labor for him, no chemical stains, no sunstroke. Those joys are reserved to the active gardeners under his guidance. Stan just takes his notes back to the computer system and lets it calculate new labors for the following day.

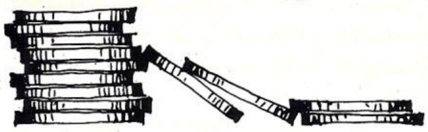
Clearly, the computer has no inherent distaste for organic gardening techniques and will happily specify mulches and insect predators in place of poisons.

Sensibly, Stan collects data on the performance of his two-dozen or so customers, so the computer can compare real experience in specific gardens with all the theory he fed to it from books. In the second year, local reality will have an influence on the calculations.

By the end of the season, Stan has received pay that works out to fifty cents an hour or so, counting the preparatory winter work. He has learned more than he really wanted to know about politics, commerce and bureaucracy, and has been presented with three or four tons of produce by his proud clients.

He has probably also identified him-

Write for Spare Change



You can try a rather conservative way to make a few dollars with a personal computer system. The method is indirect, but consider it.

You may develop experience, programs and hardware that will be of interest to other people. If you can write about these developments succinctly and easily (and here's the catch: it's seldom easy to write anything that other people want to read), you will find magazine editors eager to examine your work. If they like what they read, they'll offer you small sums of money for the opportunity to publish your work. If they *don't* like the work, they'll send it back to you.

PERSONAL COMPUTING, for example, wants to see good meat-and-potatoes material treating personal computing applications, software and systems. Our need for fiction is real but limited. Our need for "think pieces" is real but limited. We have no need at all for moralizing editorials disguised as technical articles.

If you're a writer already, you know the rules of the game and will be interested chiefly in our scale of payment, which runs from three cents to five cents a word, depending on the editorial effort necessary to make a piece work. Clean, professional work is good for a nickel a word.

A full page of text, without illustrations or advertisements, runs about 900 words.

If you're not already a writer, note that a single page of typewritten, double-spaced text runs about 250 words. Four pages of such material, professionally rendered, might bring $50 to the author. Articles run 500 to 4000 words, typically. Good short pieces are harder to find than mediocre long pieces. Shorter is better.

It's customary to send the editor the original, typed manuscript, not a handwritten draft nor a grubby carbon copy. Many editors refuse to consider Xerox copies of manuscripts, partly because the copies are often poor and partly because they wonder why you withheld the original. Have you sent it somewhere else?

Neatness counts. Clarity is of utmost importance.

It is also important to send a stamped, self-addressed envelope along with the manuscript, so the editor can send it back to you promptly without spending time and money on the effort. The SASE is your best bet for seeing your manuscript again if it is rejected. No publication guarantees to return your materials, but they *try.* Don't hold your breath. Responses from editors may take weeks or even months. PERSONAL COMPUTING will strive to be more prompt.

If you want to apply your brain and your computer to generating written material, consider getting a good text-editing system that lets you use the computer to help you type.

It's embarrassing to admit it, but writers often discover that the most harrowing part of their work is not rethinking and re-editing but re-typing.

Few of us are good typists. If you undertake Lemonade Authorship, you can improve your standard of living enormously by equipping yourself with a computer-driven electric typewriter. A good system remembers everything you write, stores it in memory, lets you edit the material to correct spellings, change words and fix typographical errors, and then types the corrected text out cleanly, automatically. Yes, you'll find several commercial systems on the market (your local office equipment dealer will tell you all about them) but they cost an arm and a leg, by non-corporation standards. Your personal computer (a general-purpose machine serving many other purposes) can be used in this task.

But, please, use a real typewriter with standard upper and lower case letters and standard punctuation marks. Teletypewriters and printers typically do a rotten job of typing text that an ordinary person is expected to read, in comparison with ordinary typewriters. Have pity on your reader and improve your odds of selling by typing clean, conventional pages of text. If you teach your computer to draw — good, clean, conventional images — magazines will be glad to see the work.

For a handy guide to the rules of the writer's game, consult some work like *Writer's Market* that not only tells you how to present your writings but points to many markets, though not many for computer material.

Think of PERSONAL COMPUTING when your Lemonade Magazine Article Service is up and working.

self to the community as the man to see for scheduling bowling leagues and organizing other bodies of low-level information so they can be used easily. The parents of other youngsters who like computers are probably offering him a little pay for helping the newcomers with their own projects.

It is likely, too, that some nurseryman has mentioned Stan and his enterprise to one of the seed or chemical companies. They may be eager to use the programs and data he has organized, or they may press him to make use of programs they have prepared commercially. In either case, he is the beneficiary of their attention.

Agreeable Profit

Stan, the nurseryman and the alert company that discovers him stand to profit agreeably from his efforts to raise local agriculture to record highs of productivity. Best of all, perhaps, Aunt Yadwiga will have been pacified so that Stan's problem will be to keep her from bragging about him with too much effusiveness.

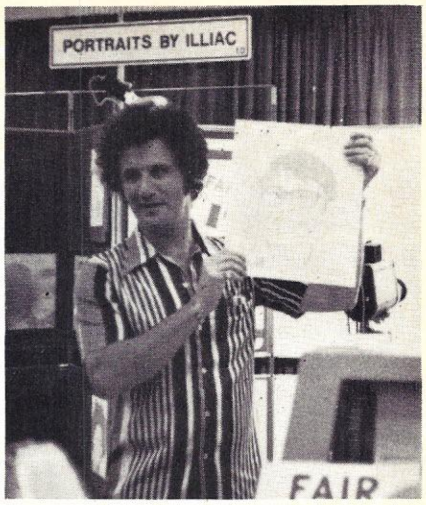
Then again, he may bollix the whole thing, antagonize the neighbors and blight community interest for anyone else with a computer. One never knows until later. That's what makes these ventures so exciting.

Odds are, though, that the average touch-of-death gardeners who use his services will be happy to see him again next spring. That's nice. Anyway, what do you want from a kid with a lemonade stand, anti-trust violations?

"What would young Stanislaus do with a computer if he had one?"

Lots of things. We'll track the LEMONADE COMPUTER SERVICE COMPANY in future articles.

"ILLIAC" THE FAMILY COMPUTER~ ENTERPISE...



Jim Small doing his act. Panasonic camera at his elbow, Printronix printer foreground.

At least half a dozen small enterprises are making the most of a Lemonade application that has become economically feasible only recently. The service: take an electronic "snapshot", print it out promptly and give it to the customer.

The pictures are appealing because they're large (the portraits of Editor Nels Winkless and Publisher David Bunnell displayed shamelessly here are 10 x 11½ inches in the original) and because producing grey tones is surprising and entertaining, if not new.

One of the more successful companies in this business is Portraits By Illiac, 213 Executive Drive, Guilderland, N.Y. 12084. Don't plan on reaching the proprietors quickly at that address, because they're usually on the road, all over the country.

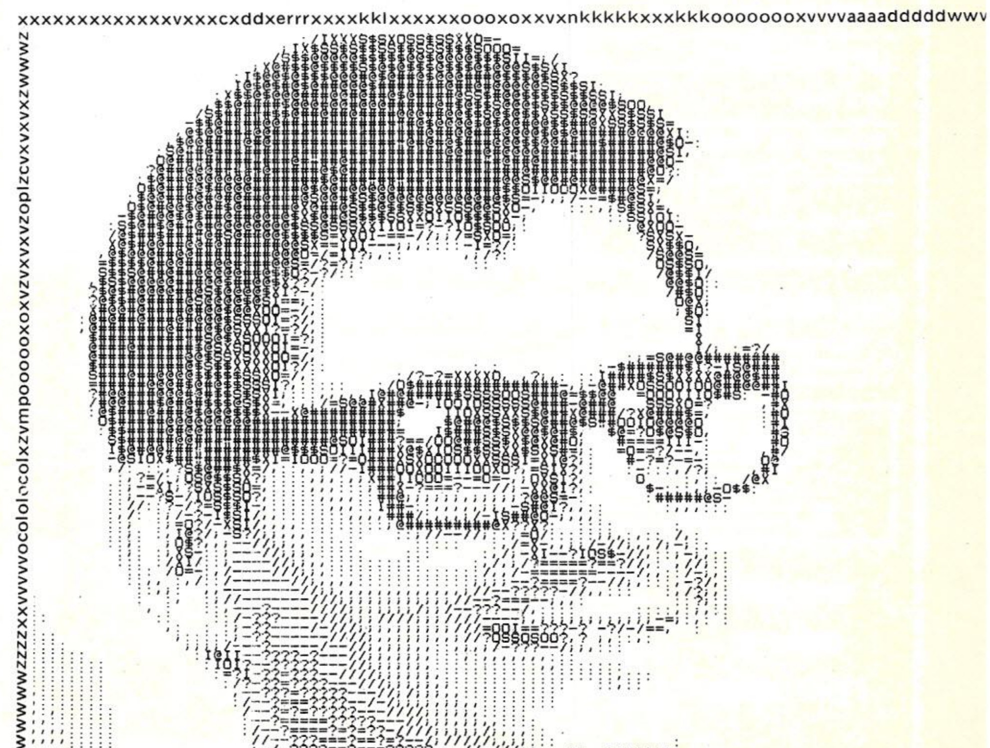
Illiac is a family enterprise. Its leader, Jim Small, is an escaped computer scientist from GE who tried his hand unsatisfactorily at the travel agency business. He then recognized the virtue of his wife's insight that computer-generated portraits were a marketable commodity. Mom, Dad and their two attractive grown kids (Dave, a high school senior, and somewhat older Lisa who didn't volunteer her age) had a special travel trailer constructed, assembled the necessary technical gear and began a gypsy life. The Smalls work at major fairs selling pictures to fascinated impulse buyers at a price too good to pass up.

We watched a typical early-evening fair crowd for nearly an hour while Jim Small took in dollar bills and passed out pictures as fast as he could. His rate? Maybe one a minute. During the extended slack times, the Small family had time to talk to visitors.

Small's capital investment was by no means insignificant. Illiac is a small but sophisticated hardware system. Its details are proprietary, but the general scheme is this:

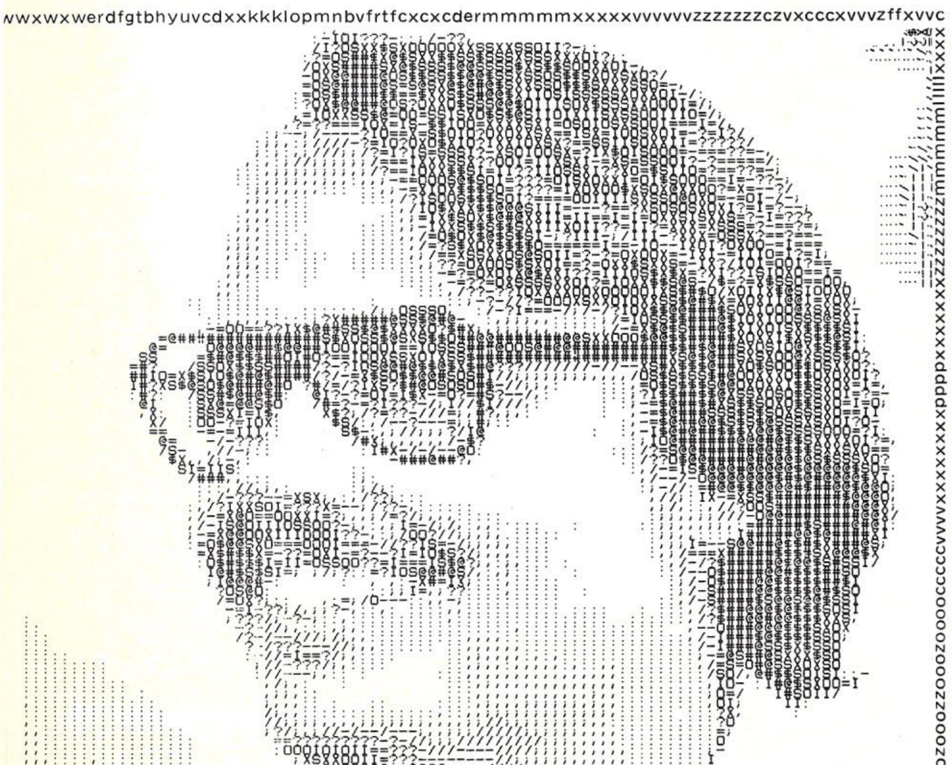
An ordinary Panasonic black-and-white television camera with a zoom lens produces a picture of the subject, who is seated on a stool and well-lighted by a pair of lamps. The television image is displayed prominently on two monitors. The operator watches one monitor to judge the pose and expression; the monitor's tube is masked with tape, so the operator sees a picture in a vertical format, like the final portrait. A larger monitor is placed prominently at the front of the booth, so the subject's friends can admire him on television and make raucous comments.

When he likes the image, the operator punches a button and one "field" of the television scan (half of the full interlaced frame) is dumped into an analog-to-digital converter of special design, digitized, then dumped into core in



Nels Winkless III

While Jim persuades the subject in the background to smile (she's partially visible on the monitor at upper right), son Dave pulls the previous subject's portrait from the printer.



the NOVA II computer that is the system's major element.

The computer examines each element of the incoming string of bits that represents the picture and determines the grey-scale of the element. "Eight grey levels," says Dave Small. That would be only three bits.

The computer then instructs a Printronix line printer to rattle out an image in which the grey level of each picture element is represented by a different printer character. A blank space represents the lightest grey tone, while the # symbol is the darkest. Dave says that 27 different symbols are used to produce the eight grey levels.

The Printronix machine is specially adapted to printing images. The spacing on a conventional printer or typewriter places the characters in a line quite closely but spaces the lines and leaves a significant gap. If you printed out an image on a conventional device, the whole thing would be greatly elongated, not proportionally correct. The Printronix device pulls the lines in closely to preserve the proportions.

The printer is also fast enough so the image of the subject is already crawling steadily up the platen behind a plastic cover, to the comments of the audience, by the time the subject has left his stool and worked his way to the front of the booth.

The operator lets the paper picture rest in the printer for a few seconds while he snaps the picture on another subject. He turns around, tears off the picture, shows it momentarily, slips it into a clear plastic envelope and trades it to the customer for a buck.

It's a smooth, theatrical operation.

The Smalls have two systems working in parallel and each reportedly cost about $25,000, more than the average hobbyist wants to plunge for.

The systems aren't quite identical. One, with its program stored in PROM, can be reloaded in a couple of seconds. The other, with no PROM storage, calls for 20 minutes of tedious switching whenever it faints from an interrupt, poor grounding or a jolt of static. Maintenance is not always easy in the middle of a carnival. Every trade has its drawbacks.

This is show business, complete with a barker's PA system to bring in the crowd.

The NOVA, a 16-bit system with 32K of memory, is housed in an impressive, irrelevant five-foot rack and fitted with lights that flash in a subdued, dignified fashion. The rack is nearly empty, of course.

The nicely made booth, neither gaudy nor junky, has a sort of showy technical effect. The Smalls look forward to taking it abroad.

The inherent problem is to figure out an encore. Really, this job doesn't need a general-purpose computer but a box of logic dedicated to the single purpose. The market seems insufficiently large to pay for such a specialized system and will inevitably taper off in interest after a few years, when everybody has at least one printed portrait on his wall. The trick is to adapt the general-purpose computer to another, equally interesting application.

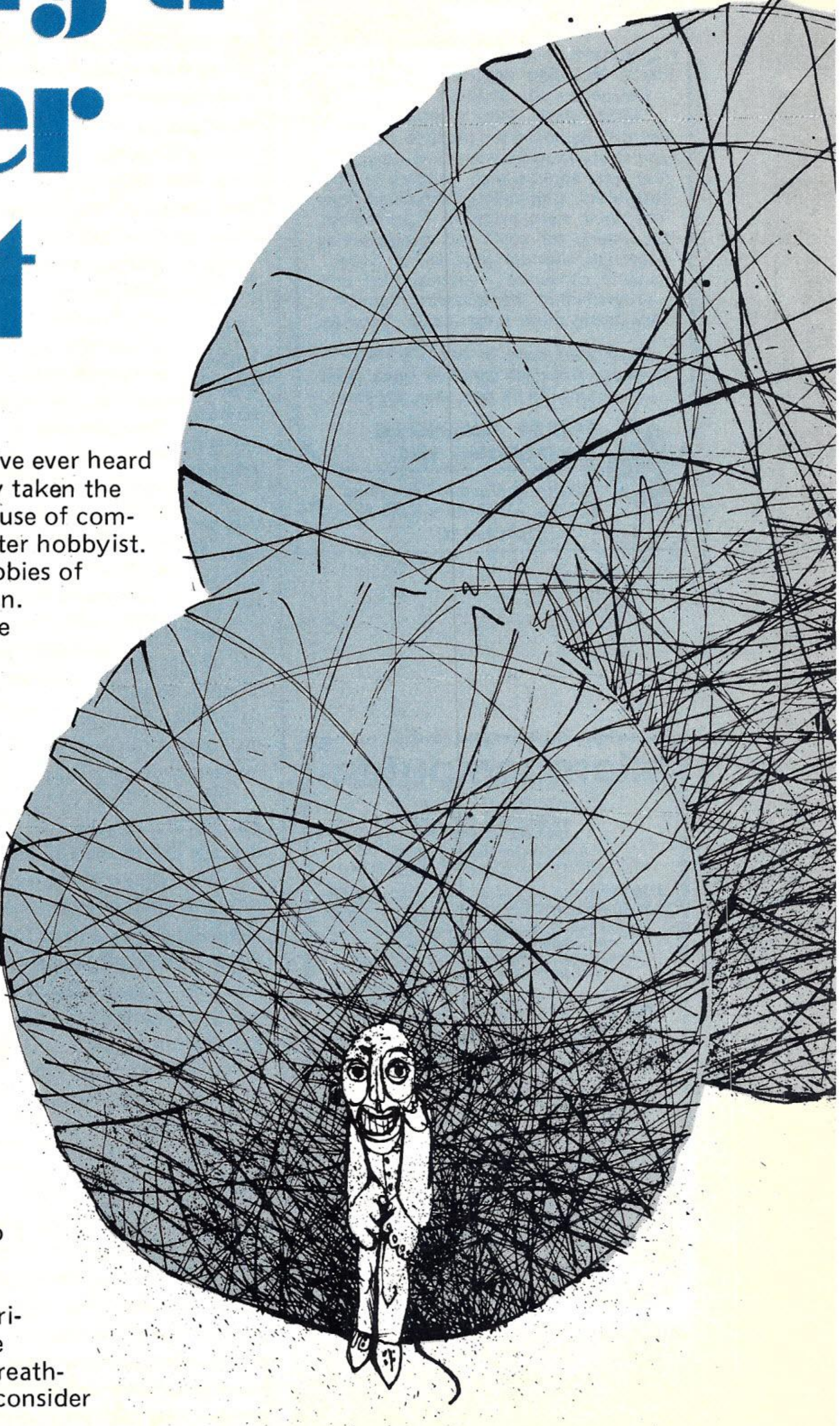
But what? Colored pictures? 3-D? Sculpture? Laser etchings?

"We're working on some good ideas," says Dave confidently. (Having run off and joined the circus with his family, he's eager to stay with it.) He doesn't hint at what those ideas may be.

Until some Lemonade entrepreneur develops a new product at a proper price, the field is wide open and unpredictable. Meanwhile, the Smalls will be putting Illiac through its paces, giving the customers what they want and enjoying the open road. Watch for them.

ten easy steps to becoming a computer hobbyist

by Henry Gilroy



Unless this article brings you the first news you've ever heard about small computers, you've probably already taken the first step toward the personal, non-professional use of computers. You are in danger of becoming a computer hobbyist.

Ours is not to recommend or disapprove. Hobbies of all sorts are pursued for good reason or no reason. The hobbyist makes his own excuses for what he does.

Newspapers used to run pictures of people who collected string. The happy hobbyist typically appeared with benign smile next to a four-or-five-foot diameter ball that had grown from a bit of twine into a monstrous spherical obsession. That ball was reassuring, satisfying in some mystical way, and the owner glowed with blissful pride.

Computer hobbyists, so far, don't present such a benign, complacent appearance. They seem harrassed and eager, with a touch of concern around the eyes, as if something may be happening that they can't quite grasp.

The string collector looks forward to something predictable — ever bigger balls of fiber. The computer hobbyist peers into the unknown, seeking the barely predictable.

Before proceeding with this article and contemplating the computer hobby, perhaps you'd like to ravel a few hundred yards of rope, just to get a feel for the broad range of hobbyism. A warm-up, so to speak.

Or if you feel you have adequate hobby experience to toughen you by virtue of model airplane work or stamp collecting, you may be waiting breathlessly to start this article. Ok, then. Ready, set, consider the first step:

Illustrations by Don Nace

step 1.

Notice computers.

Everybody is aware of computers to some extent, not necessarily happily. Ever since 1947, when early computers began to get publicity as the miraculous, mysterious mechanical henchmen of scientists, many people have had a somewhat queasy attitude toward the devices.

The queasiness was not relieved by news that before its construction the first hydrogen bomb was modeled in detail on the computer ("Ulam and Teller *knew* it would work."), nor by the persistent, true rumors that some computer systems performed in ways that their designers had not predicted and could not account for.

Computers were wrapped in a cloak of technical gibberish and surrounded largely by people who seemed either slightly cracked or coldly indifferent to human concerns, interested only in serving the special needs of their data processing systems.

The rest of us enjoyed contact with computers only through business systems that sometimes garbled our transactions and displayed no talent for answering our nasty letters helpfully.

Most important; we've been left out of the fun that goes with computers. There's been no conspiracy to deprive us of the fun of playing with the greatest toys of our half-century. Computers just haven't been ready for us.

For one thing, computers have been expensive, beyond the reach of the average interested tinkerer. For another, those who have nursed computers along to their present status have had their hands more than full trying to cope with cranky equipment and figure out how to communicate usefully with other computer folk, never mind the general populace.

Only in the past couple of years have computers of significant capacity been available to the amateur, systems that, like sports cars, give their operators a sense of power and excitement at the price of a good hi-fi system. Miracles have now been wrought in offering spectacular computer hardware to anybody who wants it, in translating computer gibberish into language the ordinary speaker-of-English can understand, and in removing the threatening mystery from computers while setting the fun loose for all of us.

The first step, then, is to detect what's going on around you and develop an itch to play with computers. Don't you long to make a powerful computing system jump when you say "Frog?" No? Then look around some more and notice the tantalizing personal computers that can respond to your wishes if you choose.

step 2.

Learn specifics about computers.

It's all very well to ask you cheerful computer-minded friends about personal computing, but until you already have some sense of what it's all about, you won't even know what to ask. Moreover, not one in ten of us can explain anything well. So you'll be wise to examine some well-done materials first.

You can do worse than to start with Norbert Wiener's works of 25 years ago, things written even before the advent of transistors, when computers were still big, hot, slow vacuum tube systems. Wiener coined the word "cybernetics" and had many prophetic things to say about the subject. Since computer jargon had not developed greatly, Wiener could write in rather ordinary language about exciting new concepts of which we are now enjoying the fruit. Your local library is full of his work. Smart man. Good stuff. Mostly philosophical.

A more recent writer gets promptly to the point of explaining in simple analogies, with words of one syllable what computers are and do. He is Nat Wadsworth, of Scelbi Computer Consulting, Inc. in Milford, Conn.

Wadsworth did such a clean job of writing a manual for a Scelbi product that his section on the fundamental operation of computers has been widely used to instruct computer novices. *Byte* Magazine republished that section as an article entitled "Computers are Ridiculously Simple" in one of its early issues. That issue is surely in the library of someone in your local computer club, and you will be well rewarded for applying yourself diligently to its easygoing discussion.

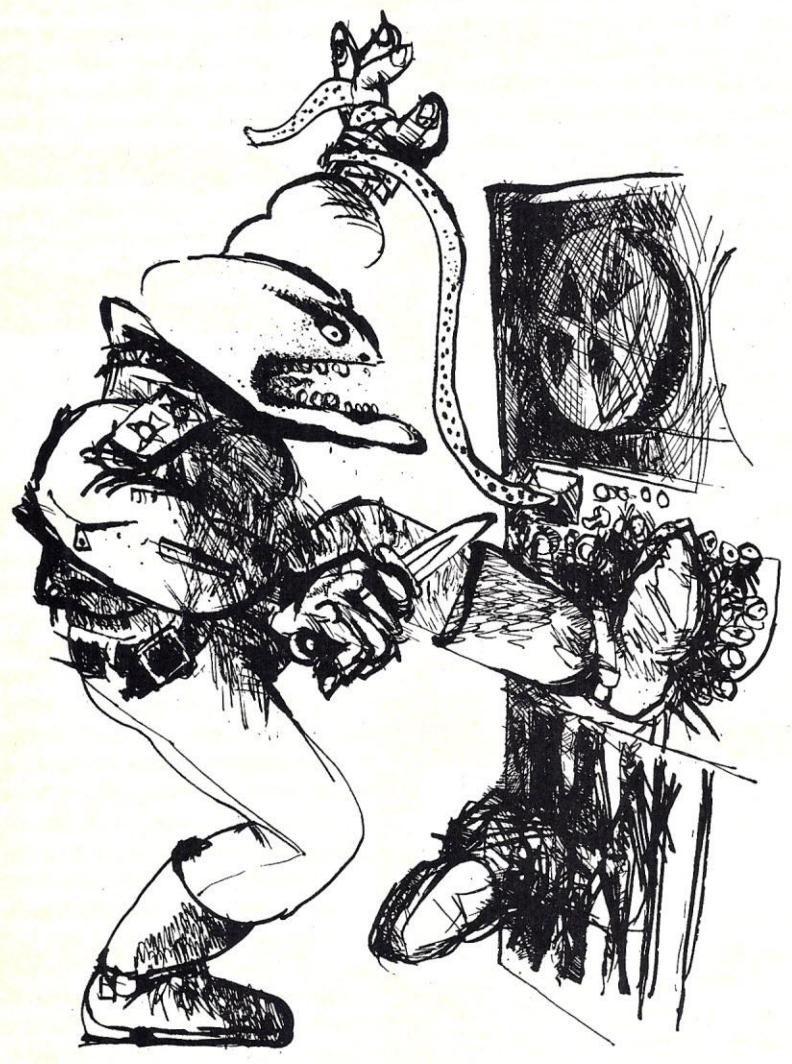
Theodore H. Nelson performed a similar service with respect to computer software in his book, *Computer Lib Dream Machines,* published in paperback in 1974. The book contains a lot more than a simple explanation of

programming languages, and you may enjoy the whole thing. It's a mixed bag of opinion and straightforward information, though, and you may not be able to evaluate it sensibly until you learn a bit more.

You can find all sorts of works, ostensibly for beginners, that explain Turing machines, Boolean algebra and techniques for optimizing mathematical equations for computer operations. If your taste runs to math and mechanical puzzles, these works may be just your meat. If you're an innocent passerby trying to cope with something new and interesting in your life, hold off on math until you are able to tie the theories to something familiar. If art and history are your usual interests, don't let anybody put you off by insisting that computers are inherently mathematical systems. That may be argued later, but you don't have to join the argument first off.

Read some of the many books and periodicals that fill the racks of the computer stores: *Interface, Byte, Creative Computing, Dr. Dobb's Journal of Computer Calisthenics and Orthodontia,* and, of course, PERSONAL COMPUTING. You'll learn some entertaining specifics here and there.

Continued



Ever since 1947, when early computers began to get publicity as the miraculous, mysterious mechanical henchmen of scientists, many people have had a queasy attitude toward the devices.

Science Fiction

We're interested in publishing short science fiction articles with mini-computer central themes. Payment (consistent with length and quality) made upon acceptance. Must be original and not elsewhere published. Submit articles with self-addressed stamped envelope to **Personal Computing**, 401 Louisiana SE No. G, Albuquerque, NM 87108.

you think you got problems

You don't. Because you've probably come up with some solutions our readers would like to hear about. You've probably come up with a new computer game. Or maybe you've figured out a novel way to interface a frammistan with a whammistan. Whatever, you ought to let your fellow computerphiles know about it. Write an article for **Personal Computing**. And write it clearly enough for a hobbyist or novice to understand. Send article, article idea or outline to: Editor, **Personal Computing**, 401 Louisiana SE No. G, Albuquerque, NM 87108. Payment varies with length and quality of article. Include stamped, self-addressed envelope.

step 3.

Find and join a computer club (or start one)

Computer clubs have sprung up all over the United States and have total membership estimated in the tens of thousands. New clubs and chapters pop up monthly.

Why join?

Maybe the camaraderie of a hobbyist's club appeals to you. You can hobnob with in-group people and learn to speak the jargon that marks you as an insider. All sorts of hobbyists tend to congregate this way, and those who enjoy the clubbiness tend also to be the dedicated people who do the back-breaking work necessary to maintain an active organization.

Even if you're a loner who looks forward to solitary pleasures in quiet communion with your computer, you may profit from club membership. For one thing, most clubs publish local newsletters of some value that report on nearby resources in hardware and people.

Further, most clubs have made commercial arrangements that allow members to purchase hardware – and some software – at significant discounts. This can amount to real money savings. Increasingly, clubs are meeting in computer stores, whose proprietors hope to win continuing business by providing facilities and services. You may gain much from this.

Chances are that you'll find company you enjoy in a club . . . and in this rapidly changing field, you'll be glad of company. *(See list of clubs on page 94.)*

step 4.

Determine if you are fish or fowl.

That is, decide whether your interest lies in *building* computer systems or only in *using* them. Yes, kits are fairly easy to assemble and almost anybody can manage it, especially with aid from a knowledgable companion. However, you may find yourself completely baffled by the requirements of the systems if you hope to add to them as you go along.

"Do you want ASCII or Baudot interfacing?" What?

"Yes, that's rated at 24 volts, but what's the RMS voltage?" Who?

If you like this sort of thing, you can learn all about it and spend the rest of your life sporting happily among handbooks of electronics and logic.

If you don't want to learn everything there is to know about connectors, power supplies and coax cables, make sure you don't maneuver yourself into the position of *having* to deal with them. You do have a choice. You can pick your equipment, your associates and your objectives to best effect if you determine your own strengths and weaknesses early in the game. Then you should accept them or change them deliberately.

This is not a recommendation to shrink from new challenges and information. You may astonish yourself with your quick grasp of electronics or programming. Just don't lock yourself into a potentially frustrating course of action until you've searched yourself for surprises and settled on some attitudes.

By all means, play with a computer yourself. Use a friend's or try a system at a store. Learn about yourself, not just the hardware.

step 5.

Decide on what you want.

The whole point is to get a computer under your own control. That means you'll want some hardware at home, and it's up to you to pick it out. Yes, it's considered fair to get some help.

Don't put too much thought into the used commercial computers you may be able to buy inexpensively. As long ago as 1969 a major electronics trade paper headlined a story that some \$3 billion worth of used computer equipment would be in the market by 1971. They were right. The sellers, rather would-be sellers, were none too happy.

Some hobbyists do cannibalize these old systems and select elements to build up new systems of their own. There's a lot for them to choose from. At one time Greyhound Corp. (which not only owns a major percentage of the commercial jetliners in service but has been in the computer leasing business on a mammoth scale) reportedly had a warehouse stuffed with 400 IBM 7090's, the wonder computers of the

early sixties. Surely this represents a treasure trove that the hobbyist may hope to exploit. No. Sorry. You wouldn't like it.

The care and feeding of a big old computer is a complex, expensive undertaking. The finicky machines demand air conditioning of extraordinary power and consistency. They require cable harnesses of staggering complexity. They interface only with certain kinds of expensive, obsolete, unavailable peripherals, tape decks, printers and card readers. They are physically big.

Worst of all, they don't give you much more computing capability than the tidy little new personal computers that sit quietly and uncomplainingly on a table or desk.

If you are an elektroniker, a craftsman, machinist, inventor and innovator, you may be able to assemble a wholly satisfactory system out of old computers. But the changes in computer technology have been truly revolutionary. Nothing is so outdated as an obsolete computer. Good used systems aren't much available in the market, certainly not inexpensively. Most computers worth having are still at work.

You may not want your own computer at all. You may want a computer terminal tied by telephone to a computer at a timesharing facility. The minimum charges for such systems these days are modest, and they provide access to enormous computers that can gladden your heart with their performance. The trouble is that the computer *isn't* truly yours.

You can't turn your back on the timesharing terminal for a month without continuing to pay minimum charges and, if you do use the system heavily, building files of programs and data in the central system, you quickly build up hefty monthly charges for storage of that material. Timesharing may be just the ticket for a business with enough employees to maintain consistent average usage rates. However, the hobbyist lives in proper dread that the unending charges will eat him up, and he'll have nothing to show for it. Do look at timesharing. It may be just right for you, though not many other amateur computer users have found it so.

Your best bet is to go to a computer store and ask the clerk for help in choosing a computer system of your own. You'll get a lot of help along with useful experience on his demonstration system.

If you are out of reach of a store, send for catalogs from the manufacturers who advertise in PERSONAL COMPUTING, for example, and curl up in a chair with the lovely pictures and text for a few evenings. Seek advice and interpretation from club members or anybody else who has direct experience in personal computing. (The guy who runs the IBM 360 down at the bank may not even know what you're talking about, so don't depend on him unless he makes the proper signals you have come to recognize in personal computing.)

step 6.

Find enough money to pay the bill.

The corollary to this instruction is: curb your appetite to suit your budget. It is very easy to get carried away and commit to more than you can handle financially. Before the stuff arrives, be sure you can pay for it. Sell your motorcycle. Hock one of your kids. Get a part-time job. Confer with your banker about the elasticity of your Mastercharge card or Bankamericard.

The economics of this hobby are important. It's in the same league financially as sailboating, motorcycling, hang gliding, flying radio-controlled model airplanes, even skiing. (Or haven't you looked at ski boot prices lately? Some boots alone cost as much as a small computer kit.) Computer prices have dropped dramatically in the past couple of years.

Take comfort in knowing that prices and system performance will change significantly, steadily in the next decade. You *will* be able to upgrade your system economically as the months and years pass. Don't hold up a liquor store to finance a giant first purchase. Just arrange for a system you can handle at practical cost and then build on that basic system.

step 7.

Brace yourself for the future.

You know perfectly well that today's new system will seem dated in a year. You may be shocked and infuriated to realize that if you'd waited a few months, you could have bought far more performance for the same money. But if you wait a few months . . . and a

few more . . . you'll never have a system of your own. You must start *somewhere*. (Are you really sorry you bought a pocket calculator before prices reached their present levels?)

Look ahead, though. Study the magazines and pump the manufacturer's people to learn what's scheduled or even anticipated. If you keep yourself well informed, you can take advantage of progress and avoid locking yourself into obsolescence. You'll need steel nerves, but that's part of the fun.

Manufacturers are already talking of 32,000 bits of memory on a single chip at moderate cost, of greatly increased speeds in microprocessors, of dazzling technological changes like those that have brought us to present capability.

Remember, though, that the peripheral hardware – the computer-controlled typewriters, printers, television displays with keyboards, punched tape readers and magnetic tape recorders – are not likely to change at anything like the rate of the electronics systems.

Sure, there may be some nice surprises, but in general, mechanical systems have been exhaustively developed and debugged over decades. Their performance may be improved by a few percent here and there, but they are unlikely to operate at ten times their present speed or drop 90% in cost. They did that sort of thing long ago, and the last few percent is always the hardest.

Your chosen supporting system won't become obsolete overnight. The system as a whole will plug away with you for years. Brace yourself for action in the future but be reassured by the knowledge that the tried and true will stay with you.

step 8.

Find a useful friend to work with.

Loner or not, you'll need help. It's a handy thing to work comfortably with an acquaintance who knows how to use a soldering iron if you don't and who can tell a resistor from a capacitor if you can't . . . and who appreciates your ability to write a clear letter to a manufacturer and do some programming if he can't.

Maybe the camaraderie of a hobbyist's club appeals to you. You can hobnob with in-group people and learn to speak the jargon that marks you as an insider.

This isn't just a matter of being part of a jolly club. This means involving somebody directly in your project and involving yourself in his or hers to your mutual advantage. You must be able to stand each other, talk easily, look forward to new things together. Many a partnership founders and makes enemies of the partners. Beer does not solve all problems.

Work on this before your system is delivered and prepare to check the packing lists together against the boxes and bags of real stuff. Much better with two.

step 9.

Get your system up and working – then use it.

All beginners are staggered by how much they don't know after the first triumphant flush of making a computer do something, anything. Once you are certain the system is up and behaving right, you can be confident that program failures can be attributed to you, your own self. It's all right to groan a lot, as long as you stick to the game and improve your proficiency. (Computer proficiency, that is, not proficiency in groaning, though that will improve, too.)

What is meant by "using" the computer? Ah, that's up to you. By the time you have worked your thoughtful way to the ninth easy step, you will surely be brimming with ideas for what you want to do with your computer, won't you?

Maybe not. A lot of people with computers don't know how to put them to specific, useful work. They fiddle and faddle with the hardware in happy rapture. (Do you know anybody who is always happily scraping and painting his boat, not sailing it?)

The hobbyist with a computer gradually tightens his grip on the modern world, increasing his satisfaction that he is not at the mercy of other folks with computers. Just being *ready*, competent and armed is apparently satisfaction of a very high order for a lot of computer hobbyists. Anything beyond that is gravy . . . and there is a lot of gravy. Lists of marvelous programs and applications may be culled from the pages of PERSONAL COMPUTING.

Do what you choose. Freedom of choice is a major part of the hobbyist's reward.

step 10.

Lure naive friends into the field.

Now that you have traveled the route from a vague perception of computers to personal competence in their construction and training, you can establish your credentials as a bona fide computer hobbyist by sharing your knowledge with a friend who hasn't yet even taken a hesitant first deliberate step. If you can really help him along, you've proved yourself.

Ten easy steps.

Well, easy looking *back* on them . . . and good fun working up through all ten.

Now, sneak up on somebody else and shove this article into his hand.

Timesharing?

Personal Computing examined this negative view of timesharing and wondered if a case might be made for the opposite view, that timesharing is a good approach to nonprofessional, hobbyist computing. We contacted one of the major companies in the field and invited a sales presentation on behalf of timesharing. Top management was cordial and promptly assigned somebody to search the corporation for the information we were seeking, after a bit of preliminary confusion and explanation of the general field of personal computing. Indeed, the company was genuinely interested in the search.

Its report, after a few days of internal discussion, confirmed the initial impression that the price firmly discourages the casual timesharing user *at this time.*

The economics may change with the passage of a few years. The timesharing companies would be pleased to find some way to address a new national market with products and services in an appropriate price range that lets them make a profit. But how, relistically?

Perhaps some of the smaller timesharing companies have already found the key to this, and PERSONAL COMPUTING would like to hear about it. Meanwhile the observation in this article seems sensible with respect to timesharing.

Sign Up a Software Scribe: The Power of a Club

People who like to document programs they have written are extremely rare. Documentation tends to be hateful drudgery, harder and less fun than writing the program itself. Ugh.

On the other hand, since the documentation may be as valuable as the program itself, your great programming work may go to waste unless you buckle down to the task of making it generally useful.

Every computer club is likely to have one specimen of the rare documentation-lover. This useful creature can be cultivated, flattered, bribed and encouraged with highly practical results. He may be persuaded to become the club documentarian and not only encouraged to prepare documentation himself but empowered to coerce other club members into preparing documentation for a central club library. Various sanctions against reluctant documenters are possible – withholding beer, modest fines, refusal to list the offender's programs in the library, public whipping.

The documentarian may also usefully teach documentation techniques and lend a hand to novices mired in their work. After a few months, the documentarian will become a popular hero whose work is much admired.

In his idea-and-data-laden work, *The Secret Guide to Computers,** Russ Walter offers a brief, useful guide to documentation as follows:

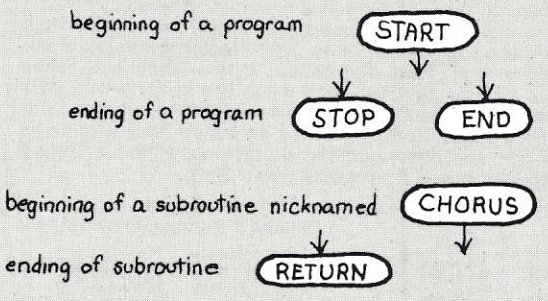
Write an explanation that helps other people understand your program. An explanation is called *documentation;* when you write an explanation, you are *documenting* the program. The documentation can be written on a separate sheet of paper (to be put in the computer center's library) or printed when the user types RUN or LIST. A popular device is to begin the RUN by making the computer ask the user: DO YOU NEED INSTRUCTIONS?

Two kinds of documentation are needed: how to use the program and the method by which the program was written. Your explanation of how to use the program should include the program's name and how to get the program from the disk; the program's purpose; a list of other programs that must be combined with this program to make a workable combination; the correct way to interpret the output; the program's limitations (input it can't handle, error messages that might be printed, round-off error); and a list of bugs you haven't fixed yet.

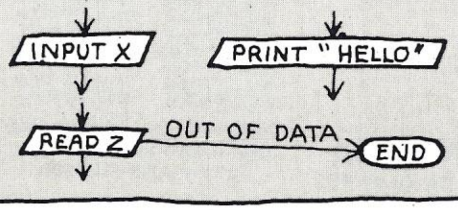
An explanation of how the program was written will help other programmers borrow your ideas and expand your program to meet new situations. It should include your name; the date you finished it; the computer you wrote it for; the language you wrote it in (probably BASIC); the name of the method used (example: this program solves quadratic equations by using the quadratic formula, solves triangles by using the law of sines, solves simultaneous linear equations by using Gauss-Jordan elimination and alphabetizes the data by using a bubble sort); the name of the book or journal where you found the method; the name of any program you borrowed ideas from; an informal explanation of how the program works ("It keeps looping until A↑2 is greater than 2*B, which makes it jump to subroutine 1000 and compute the weather forecast for Tuesday"); the purpose of each module; the meaning of each variable; the meaning of arriving at a line (for a program saying IF X<60 THEN 1000, you might make this comment: "Arrival at line 1000 indicates the student flunked."); and a flowchart.

The American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) urge you to use the following flowchart symbols:

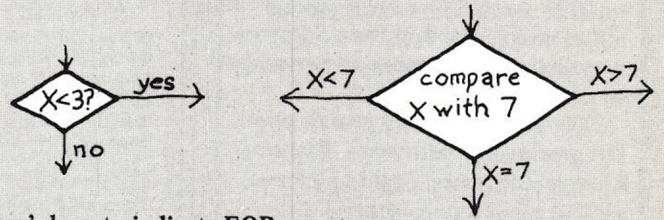
For beginnings and endings, use an oval:



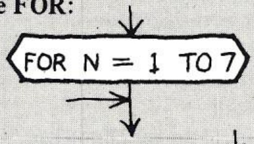
For input and output, use a slanted parallelogram:



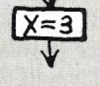
For a decision, use a diamond:



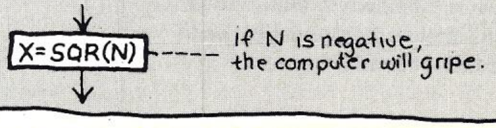
Here's how to indicate FOR:



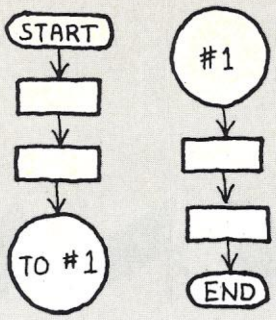
For other operations, use a rectangle:



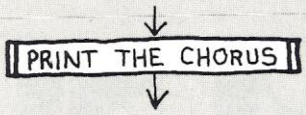
Here's how to attach a remark:



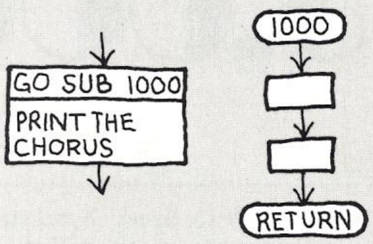
You can split a large flowchart to form smaller ones:



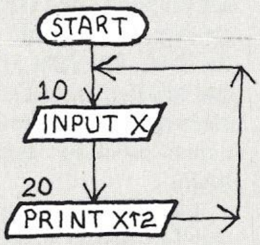
When an operation is performed by a subroutine, use stripes. If the details of the subroutine are not included in the flowchart, the stripes are vertical:



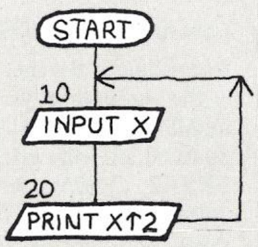
If the subroutine is included in the flowchart, use a stripe that's horizontal:



You can write the line number on the left corner of each outline:



If a flowline points in the same direction as you read English (from left to right, from top to bottom), **you can omit the arrowhead:**



** Copyrighted. Following volumes available from Russ Walter, 92 Saint Botolph St., Boston, Mass. 02116: Basic ($1.75), Applications ($2.50), Languages, Systems and Commentary (to be released).*

PERSONAL COMPUTING CONFERENCE DRAWS 5,000

Personal computers ranged from $245 to $25,000. T-shirts displayed computer love and fear. Star Trek fans zapped Klingons endlessly. Seminars taught how to choose, build and use your personal computer. The 80 booths at the Personal Computing Conference offered handy-dandy tool kits, computer art posters (free!) and a dazzling array of hardware and software. The largest personal computing conference ever, it drew about 5000 people to Atlantic City the weekend of August 28. Repeat performances are coming to other cities.

It was a bargain-hunter's paradise: most prices were slashed 10% or 15% as a "Personal Computing Conference special." Competing against established companies that have been around since the Personal Computing revolution began 20 months ago were many newcomers, trying to make a quick but honest buck by producing the same stuff a little better and a little cheaper. Many in the crowd, which included numerous 15-year-olds, plunked down about $500 apiece and took the computers home with them. The lucky ones got their computers free: over 40 companies supplied door prizes.

IBM, which had closed its eyes to the Personal Computer market, moved in to capture some of the cookies. Spurning the microscule booth it had been allotted, IBM hung overhead a little sign, luring the passer-by to go upstairs to room 1018, a lushly decked-out lounge featuring refreshments and the IBM Personal Computer, which was to the other computers what a Rolls Royce is to a Volkswagen. Priced at $8975 and up, it was accompanied by the happy face of Bruce Lomasky, who had bought one for his home, to "play games" (see separate box).

To add spice, Digital Equipment Corp. used an LSI-11 microprocessor to run a model railroad. A Boy Scout troop marched in, gathered round the trains and posed for pictures.

A rumor floated around the convention hall that Hugh Hefner sent his henchmen there to buy some microcomputers, put Playboy Bunny stickers on them and sell them at Playboy resort toy stores for playing kiddie (and other?) games. John Dilks III, one of the managers of the conference, admitted that Playboy was there but only to write a feature story, "like they do for hi-fi."

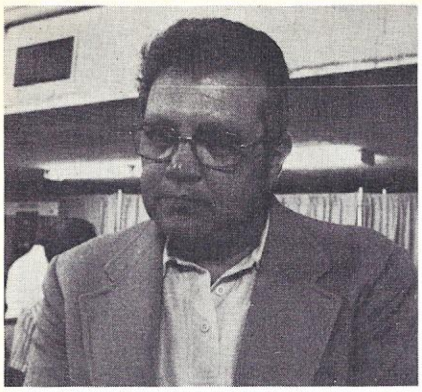
The seminars covered a wide range of topics such as "Computer Games", "The Meaning of Computer Liberation", "Making Big Bucks for the Computer Hobbyist", "Computer Synthesized Music from 0000 to FFFF", "Software for Speech Synthesis", " Comparative Analysis of Microprocessors" and "Converting Hardware Designers to Programmers". Several analyzed the relationship between Personal Computing and HAM radio. Tutorials explained specific microcomputers: the Z-80, F-8, IBM 5100, Imsai, KIM and MMD-1. In one seminar, Richard Moberg explained how he's implanting a microprocessor inside the human brain.

Carl Helmers, David Ahl, Bob Jones and Ted Nelson spoke at a filet mignon banquet where Sol Libes (of the New Jersey Amateur Computer Society) was declared "Computer Hobbyist of the Year". As an after-dinner treat, John Whitney showed his computer movies.

Love and Fear on T-Shirts

According to the most popular T-shirt at the conference, you should TAKE A COMPUTER TO LUNCH. Another pictured an evilly grinning FORTRAN FREAK. *Creative Computing* (Box 789-M, Morristown, NJ 07960) sold them for $4 apiece, along with shirts picturing the Starship Enterprise, a Bionic Toad and a computer-generated Blocpix of Albert Einstein under the words CREATIVE COMPUTING.

N.B.C. M/C Imports (Box K, Mays Landing, NJ 08330) undercut Creative



George Willbanks
Washington, D.C.

Computers are my profession. We deal in large computers. I'd like to fool with them at home. I'm getting ready to do an extensive home system — like a Zilog Z-80 with two floppies and a parallel CRT, having color and character generators in RAM, so you can use graphics. I can probably buy the whole thing for about $4500.

What will you do with it?

I have several applications that might work out, but they're not important. It's for my own satisfaction. At work I cannot touch the equipment; at home at least I can fool with it.

David Hadden
Army

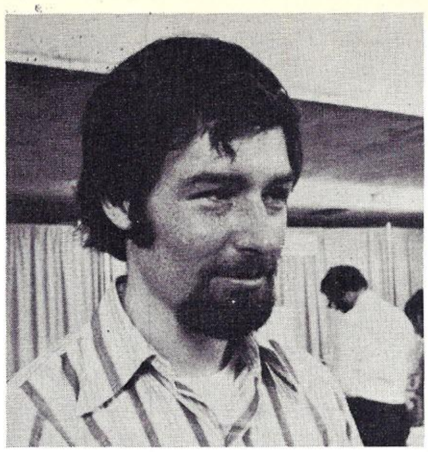
I work for the Army — research and development in computers. I have a home computer: a lot of our recent work has been with microcomputers, particularly an 8080-based system.

You're doing Army-related work at home?

No, the work at home is for fun.

What do you do with it?

It isn't done yet. My primary interest is games. Some word processing, also: the generation of record files, editing, correcting, and line-justification, because I don't type very well. It would be handy to type my letters and have them come out neat.



Tom Hughes
New York City

I have just a terminal, but I'm saving money for a home computer.

Are you doing anything with your terminal?

I get into timesharing.

Are you timesharing into a network?

It's a gray area I'd rather not get into.

You'll be using the microcomputer for the same application?

Yes, I'll get off the network once I get my system going.

What kind of job do you have?

I work for the phone company.

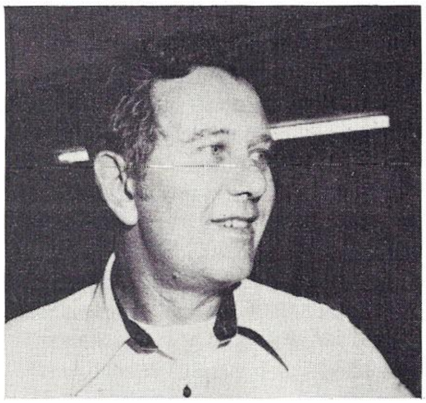
Computing's prices, by offering T-shirts for $2.75 (white) or $3.25 (color). The shirts urged you to DO IT WITH A COMPUTER, emblazoned COMPUTER LIB on Ted Nelson's fist, hinted that COMPUTERS DO IT THE BEST and proclaimed PERSONAL COMPUTING 76.

Did you ever wish you were a computer? *MITS* (2450 Alamo S.E., Albuquerque, NM 87106), manufacturer of the well-known microcomputers Altair 8800b and Altair 680, offered a T-shirt to make you look like an Altair. It has an on-off switch, positioned at ON, and says you are a MITS ALTAIR, THE AFFORDABLE COMPUTER. Four bucks.

The Hardware Hustle

Since MITS is the largest manufacturer of hobby computer systems, the MITS people came with the largest booth, most literature and most operating equipment. Besides the Altair 8800a, they displayed the new Altair 8800b with dual floppy disks, line printer and CRT. Their Altair 680b, which contains a Motorola M6800 CPU, came with a whopping 33K memory and was running BASIC on a Teletype.

MITS showed off its new Altair 7000 Graphics Printer. This electrostatic printer can also be used as a plotter and for graphics. Software lets you generate any size or style of type. The 7000 costs $785, with delivery in 60 days. It is limited by a 4-inch-wide print area and needs more software.



Ray Borrill

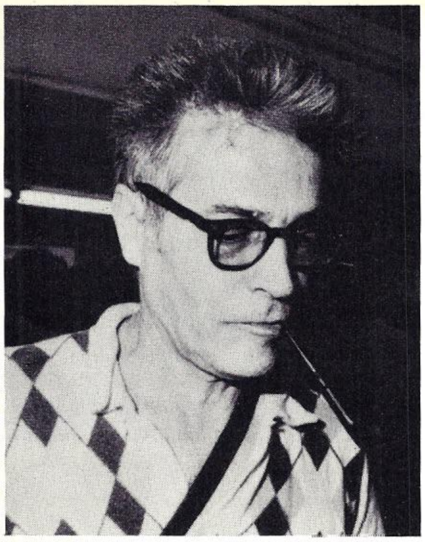
While MITS continues to be the biggest with the mostest, Atlantic City demonstrated that the "IBM of small computers" has plenty of competition. Many dealers at the conference were pushing instead the Imsai 8080, manufactured by *IMS Associates* (14860 Wicks Blvd., San Leandro, Calif. 94577).

Ray Borrill, an Imsai dealer who heads a computer store called the *Data Domain* (111 S. College Ave., Bloomington, Ind.), claimed the Imsai 8080 is about as good as the Altair 8800b and much cheaper. But he avoids Imsai's peripherals, which he says are priced too high. He makes a good profit on customers who want to buy Imsais already assembled, since he pays only $100 to assemble them.

He believes the market for Personal Computers is changing from the "tinkerers" armed with soldering irons to the new crowds who want to buy already assembled computers, complete with operating systems and ROMs, and preferring turnkey systems over meaningless blinking lights.

Marty O'Boyle (M.J. O'Boyle & Associates, Box 9094, Pittsburgh, Pa. 15224) has been running a bargain mail-order computer store for the past three months. He'll sell you an Imsai 8080 at $100 below list price. He also features a 9" CRT that's plug-to-plug interchangeable with a Teletype, prints 65 characters per line, handles speeds up to 9600 baud and costs only $495.

Although the three most popular microprocessors among professional engineers are the Intel 8080, the Motorola 6800 and the Fairchild F-8, the hobbyists at Atlantic City showed little or no interest in the F-8, because it lacks BASIC; but *Veras Systems* (139 Hampshire St., Cambridge, Mass. 02139), a company that began last



David Zernoske
Airwaves

I own an Altair 8800. So far, I've just got the basic unit together. I'm taking a course in assembly programming. I want to connect a hand-held calculator to a microprocessor, to combine the compactness of the hand-held and the memory of the microprocessor.

Do you have a mathematical job?

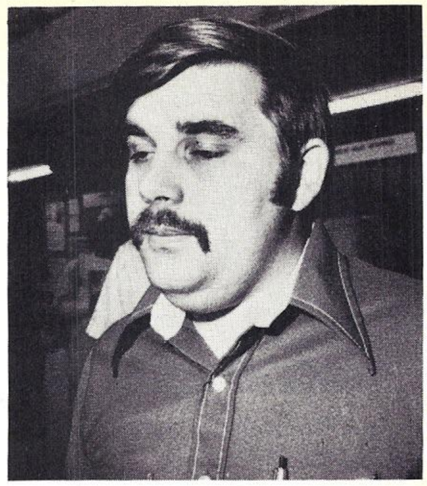
I'm an engineer at a radio station.

Dave Fry
IBM

I read several magazine articles about how tiny computers had been gradually taking over, especially for home use, so I decided to see.

Are you planning to buy one?

Just curious. I work with large computers, not small ones. I work for IBM with the large 370. I didn't realize there were so many small computer makers.



Mike Shebesta
Cincinnati, Ohio

I'm a public accountant and teach accounting at the University of Cincinnati. I'm here to develop my own business accounting system. I came to find one of the Zilog Z-80 OEM systems, a one-box dual-disk-drive type.

What will you do with it?

I have small business clients. I want it to handle my record keeping, so I can function more as an accountant than as a bookkeeper.

January, promises to write an F-8 interpreter for tiny BASIC by October 15, and full BASIC by December 15.

Hobbyists at the convention were more interested in the 6502 microprocessor and its incorporation into the KIM-1 computer system manufactured by *MOS Technology* (950 Rittenhouse Rd., Norristown, PA 19401). The KIM-1, fully assembled and tested, costs only $245 (plus $4.50 for shipping and handling), making it easily the cheapest microcomputer system on the market. It includes the 6502 microprocessor, a 1K RAM, a 2K ROM containing a monitor and operating system, 30 I/O pins, two interval timers and a six hexidecimal-digit LED display. The only thing missing is the power supply. If you want fancier I/O than the LED display, you can easily attach a terminal: just plug in four wires. The terminal interface is already included on the KIM-1 chips, and so is an interface for audio cassettes. But few computer store dealers stock it, because MOS offers only 4% discount, whereas Imsai offers 25%.

The *Hoboken Computer Works* (Box M1055, Hoboken, N.J. 07030) offers the novice hobbyist six weeks of instruction (an introduction to all small computers) plus a KIM-1 with an attached power supply, all for $389. For an additional $110, you can bring your husband or wife or friend to the classes with you. The Hoboken Computer Works is keeping its profit margin low in hopes the graduate will get "hooked" on computers, want to buy a more expensive model and return the KIM-1 for a trade-in.

The competition extended down to the board level. The Cybercom division of *Solid State Music* (2102A Walsh Ave., Santa Clara, Calif. 95050) offered an 8K RAM kit for $250, minus a $25 convention discount. If plugs into an Altair 8800 or Imsai 8080, but costs less than the memory provided by MITS, Imsai and other competitors.

For hobbyists who have more money to spend, *IBM Corp.* offered its luxurious 5100 Portable Computer. The minimum configuration costs $8975, weighs 46 pounds, is completely assembled and includes a 5" CRT having 16 lines of 64 characters, a switch to alternate between black-on-white and white-on-black, a keyboard including BASIC command words, a 200K tape cartridge with drive, a 16K RAM, 4.4K of which are reserved for the BASIC interpreter, and ROM holding more of the BASIC interpreter and the operating system. To replace BASIC by APL (*A Programming Language*) costs $1000; to replace BASIC by a dual BASIC/APL system costs $2000. In the dual system, you alternate between BASIC and APL by just flipping a switch on the console. If you want a printer, additional memory, additional cartridge drives and software, you can pay up to $25,000. The microprocessor's internals are a carefully guarded secret; assembly-language programming is prohibited.

If you'd rather have a minicomputer than a microcomputer (because you'd like 16-bit words instead of 8-bit, 400 op codes instead of 80 and much more extensive software) but don't have enough money, consider the PDP-11/03 manufactured by *Digital Equipment Corp.*(146 Main St., Maynard, Mass. 01754). By using firmware, it duplicates the instruction set of the popular PDP-11/40 and costs much less, though it runs slightly slower. For $2900 plus the cost of a terminal, you can solder together an 8K-word (16K-byte) version, complete with fans, power supply and cabinet. (Rather than buying the kit from DEC directly, it's cheaper to buy from resellers such as Synchro-Sound Enterprises in Hollis, N.Y.) For $10,450, DEC will sell you the system assembled, plus a high-quality CRT, dual floppy disks and the RT-11 operating system; BASIC is an extra $750; an on-site maintenance contract is an extra $110/month.

Mike Behar
Orange, Connecticut

I'm Manager of Systems Development for a consulting organization. I'm planning to buy a machine.

What will you do with it?

Develop games for my 8-year-old daughter. I'll develop the software myself, if I can't find what I'd like. I'll program in assembler.

Any other applications?

Oh sure — budget, keeping track of personal finances — but I don't want to put that kind of effort on it yet. You get to the heart of things earlier in trying to develop games than in commercial applications.

What do you mean by "the heart of things"?

Syntax analyzers, string manipulation, file techniques, and data structures. They are harder than the straight computation, such as multiplication routines and floating point.

Russell Almond
Dresher, Pennsylvania

I've done programming, but only on timesharing.

Has anything at the show surprised you?

The number of Star Trek games. Half the people running a computer system have a Star Trek game on it.

Did you come with your parents?

With my father and a friend.

Where did you have experience with a terminal?

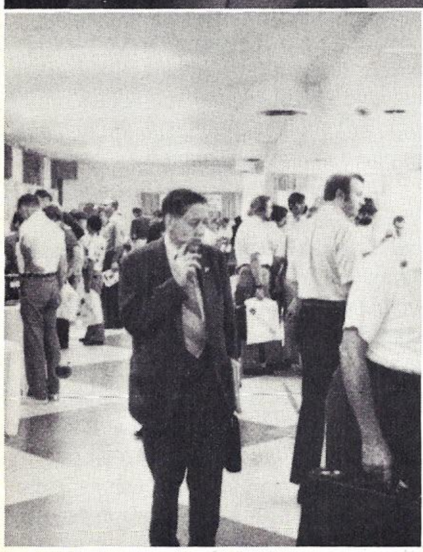
I belong to an Explorer post, which is interested in computer programming. A corporation has donated access time to us. They give us a free charge number, so we can work in the timesharing system. We have our own file. It's filled with miscellaneous computer games.

Brian Loofbourrow
Westfield, New Jersey

I'm 15. I want to get information about what to add to my computer. I have a Southwest Technical Products 6800 system.

What do you use it for?

It's a demonstration system, because my dad and I are dealers for Southwest Technical. I'm planning to put on a parallel interface, and if I can possibly slow it down to 10 pulses per second, use it as an automatic Touch-Tone dialer with memory. You'll type in someone's name, the dialing system will come out different lines in the parallel interface, and it will automatically dial the number at a much faster rate than you could normally, and without your looking it up.



Moments of fun, moments of thought.

The Software Scramble

TSC (Technical Systems Consultants, Box 2574, W. Lafayette, Ind. 47906), a new company that began on April Fool's day, displayed an ad that blasts, TIRED OF THE 'SOFTWARE VACUUM'? NEED SOFTWARE TO FEED YOUR 6800? WE'VE GOT IT! The company offers Hangman and Craps for \$3.25 apiece, Mastermind for \$3.00, a 4K slightly limited Star Trek for \$10, a floating-point package for \$5, a 60-byte random-number generator for \$1.50 and microBASIC (integer-only BASIC) for \$15.45. You get a sheet of paper that includes the machine code, fully commented assembly code and instructions. If you're too lazy to retype the code on to your computer, you can also buy cassettes. The software works on any computer using a Motorola 6800 microprocessor, such as the Altair 680. TSC is starting a Program of the Month Club that allows you to buy programs at special discounts.

To get microBASIC, you don't have to spend as much as \$15.45. TSC admitted you can get cheaper but less flexible versions from others. *Tom Pittman* (Box 23189, San Jose, Calif. 95153) offers microBASIC for \$5; his version requires only 2K instead of 3¼K. Called TINY BASIC, it runs twice as slow as the TSC version and includes only a third as many commands (subscripts and FOR . . . NEXT are prohibited); he declines to disclose the source code.

In the newsletter of *Southwest Technical Products* (219 W. Rhapsody, San Antonio, Tex. 78216), Bob Uiterwick published a version of microBASIC that has more commands than TSC's. But his version runs slightly slower and prohibits you from jumping out of a FOR . . . NEXT loop and then jumping back in.

Ted Nelson, head of the *Itty-Bitty Machine Co.* (1316 Chicago Ave., Evanston, Ill. 60201), whose initials — IBM — are not to be confused with another well-known company, is planning a new computer language called

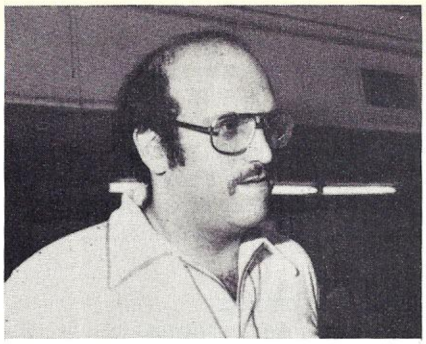
Walt Freitag, Jr.
Dresher, Pennsylvania

I'm 15. I recently won an Altair at the '76 NCC Student Computer Fair in New York, but I need a terminal. Before the fair I was only into programming. Now that I have a computer, I'm trying to get into hardware.

How do you feel about this show, compared with the NCC?

I like this show better as far as the exhibits go, because it's more geared to hobbyists and, hopefully, beginners like me. The NCC was mostly for professionals or people who would buy a thousand terminals.

Someone you interviewed — a very little kid — Russ Almond — I know him. He's also into computing. We belong to Explorer post 556, which does programming, specializing in computer games. We have a very good collection of games on perm file.



Barry & Debbie Schreiber
New York City

B: I'm here to see what they have in personal computers. I'm not a hobbyist now.

D: I'm just joining my husband.

Are you involved with computers?

B: Yes, I'm a systems engineer from IBM.

D: We have friends who own a personal computer, so we were curious. The wife has recipes on it. They play games. Their children know how to work it. They enjoy it, but it's really not our speed yet.

Have you had a chance to use their computer?

D: No, the first time I've worked with a computer has been right here at the show: my husband showed me how to play Star Trek.

FUNTRAN which will be an improvement over SMALLTALK. By Christmas he hopes to have a text editor called UPDOC, which will allow the user to insert marginal notes. The text will be stored internally by using a data structure he calls a "half ladder". He predicts 5 million homes will contain personal computers by 1980; to prepare for the onslaught, he is already marketing TEMPUS to help you plan your time and FUDGET to help you budget your money.

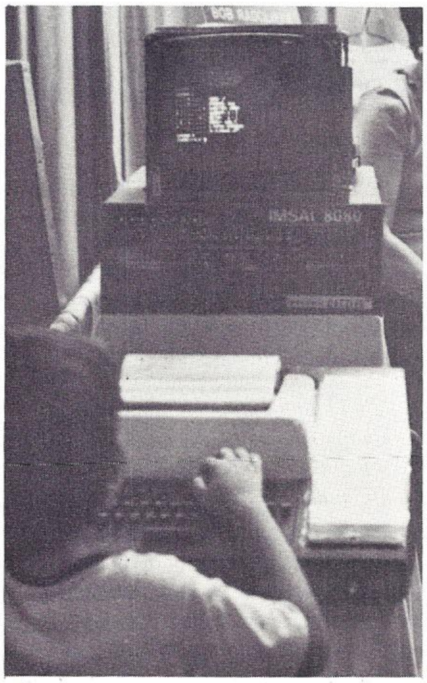
I/O Bargains for Tinkerers

You can buy a quality hard-copy terminal for only $400, from *Jeff-Tronics* (Box 81421, Cleveland, Ohio 44181). It is a souped-up IBM Selectric Typewriter, has a wide carriage (it can write 13 inches across and hold paper up to 15¼ inches across) and includes a paper tape punch and photoelectric paper tape reader. Since it prints 15.4 characters per second and uses an 8-level BCD code with odd parity, you must put an interface between it and your personal computer.

You can buy a gigantic CRT for only $279 from *AST/Servo Systems* (20 Republic Rd., No. Billerica, Mass. 01862). The 17" screen holds 41 lines of data. In addition to the standard keyboard, you get 28 special keys, 21 special pushbuttons and a gigantic power supply. Altogether, you get 210 pounds of hardware. According to Walter Reichman, vice-president of AST/Servo Systems, Burroughs manufactured the terminals for TWA (Trans-World Airlines). Because delivery was late, TWA dropped the contract. At the time of the show, he had 520 terminals to unload and was charging $495 for unchecked cartons, but now the price is even lower and the cartons are checked. The terminal outputs 52 characters per line, BCD serial.

Computers Are Magic

The Fantasy Factory (207 Abbott St., E. Lansing, Mich.) sells tricks for magicians and toy soldiers for war-gamers. The owner, Steve Senzig, intends to sell computers also because, he says, they are magic: people use computers without understanding their inner mysteries and treat them with awesome reverence as if they contain "little demons." He predicts computers will



be used mainly as expensive, magical toys or status symbols.

He claims computers can help magicians. A computer attached to a Votrax (a programmable box that produces sounds resembling human speech) can help a ventriloquist. Hiding a small Votrax inside a skull, a magician could make the skull announce which cards are in your hand. He seeks more elaborate ruses. Any ideas?

The Death of Computer Film

Attending the conference was the famous gray-haired grandfather of computer film, John Whitney. He sees computer artists deserting film in favor of videotape.

Film requires expensive silver and 24 hours processing. Videotape is cheaper, gives immediate feedback and is easily programmable because it is based on computer-compatible magnetism. Other magnetic media that fascinate computer artists are videodisks, magnetic bubbles and solid-state methods, but their technologies are not yet as perfected as videotape's.

He decried the lack of progress on videodisks (disks the average home-owner can cheaply and simply attach to an ordinary television set). Two years ago, RCA and MCA Universal said their competing units would be available by last fall. Now they say: not until *next* fall. Whitney believes videodisks are a futile dream. The arm's sensor must be able to fit in a very narrow groove — a tenth as wide as the groove on a phonograph record — but no one has yet made an arm that sensitive cheaply. **—RW**

What Your Wife Doesn't Know . . .

An interview with Bruce Lomasky

Why did you buy the IBM 5100?

I won't tell my wife, but the main reason is to play games. I was looking at the Wang and thinking about an Altair and the Tektronix. The Tektronix wasn't out yet. I ruled out Wang because I wanted something more portable, wanted APL and didn't like Wang's cassette drive. Mine has the 3M data cartridge, which is more reliable and about ten times faster.

An Altair with a floppy disk was about the same price as the IBM and more flexible: it had the floppy disk and assembly language and could interface to more. But the big thing I liked about the IBM computer is: it's small and portable. If I'm going somewhere, I can bring it with me. It has everything on the ROMs, so when I power up I have the BASIC or APL right in there. And it has the IBM support.

Do you have a service contract?

I didn't want one, but it's a necessity. When I first had it, I had a few bugs, and the servicemen were out there instantly to fix it. With an Altair I'd be doing most of my own fixing.

How much is the service contract costing you?

\$85 a month. That's with a 32K system having APL and BASIC. It's an expensive insurance policy; it's the most I'll ever pay. With that I get all the engineering updates.

Did you write the games yourself?

One or two I got from IBM. I've given them more than they've given me. I have a very sophisticated Star Trek in there – Blackjack, Hangman – 30 to 35 games. Plus I'm doing some software work; I just finished a ledger package for a bank that purchased a 5100. I have a couple of biorhythm games and checkers. I'm going to put in backgammon. I'd like to put in chess also.

Do a lot of your friends come over and play these games, or is it mainly a personal thing?

More a personal thing. A few friends like to come over. One loves the Star Trek game, and someone else comes over to see Biorhythms each day, and somebody else likes to play hangman, and somebody else wants to play word games. Blackjack's popular, because you can play it with up to 12 people. So we'll get a crowd, and everyone likes that.

Can you say more about why you preferred this to the Wang, which is about \$4000? How much did you pay?

With the APL, it's about \$11,000. With the Wang, you're talking a basic 4K; mine comes with 16K. To add on more memory with the Wang you're up about another \$1,500. I was getting the better cassette drive, a more portable unit and more expandability, because the Wang will only go up to 32K. I didn't think I'd need more then, but now I'm up to 32K, and I'm looking forward to when I can afford 48 or 64K. Then I'll rule the world.

But it was a big decision between the two because the Wang definitely was cheaper. And a big advantage Wang has is: it has more software available. The Wang BASIC is definitely superior to the IBM BASIC, which is a little weak.

At what points is it weak?

IBM doesn't have multiple statements per line, so it's slower and uses up more memory. It has a string function for character manipulation, but not the LEFT\$, MID\$, RIGHT\$ or LEN. It's confusing. For instance, if I have my last name ("LOMASKY") in there and want to add "BRUCE" in front of it, I have to put "LOMASKY" in a new variable, with blanks at the

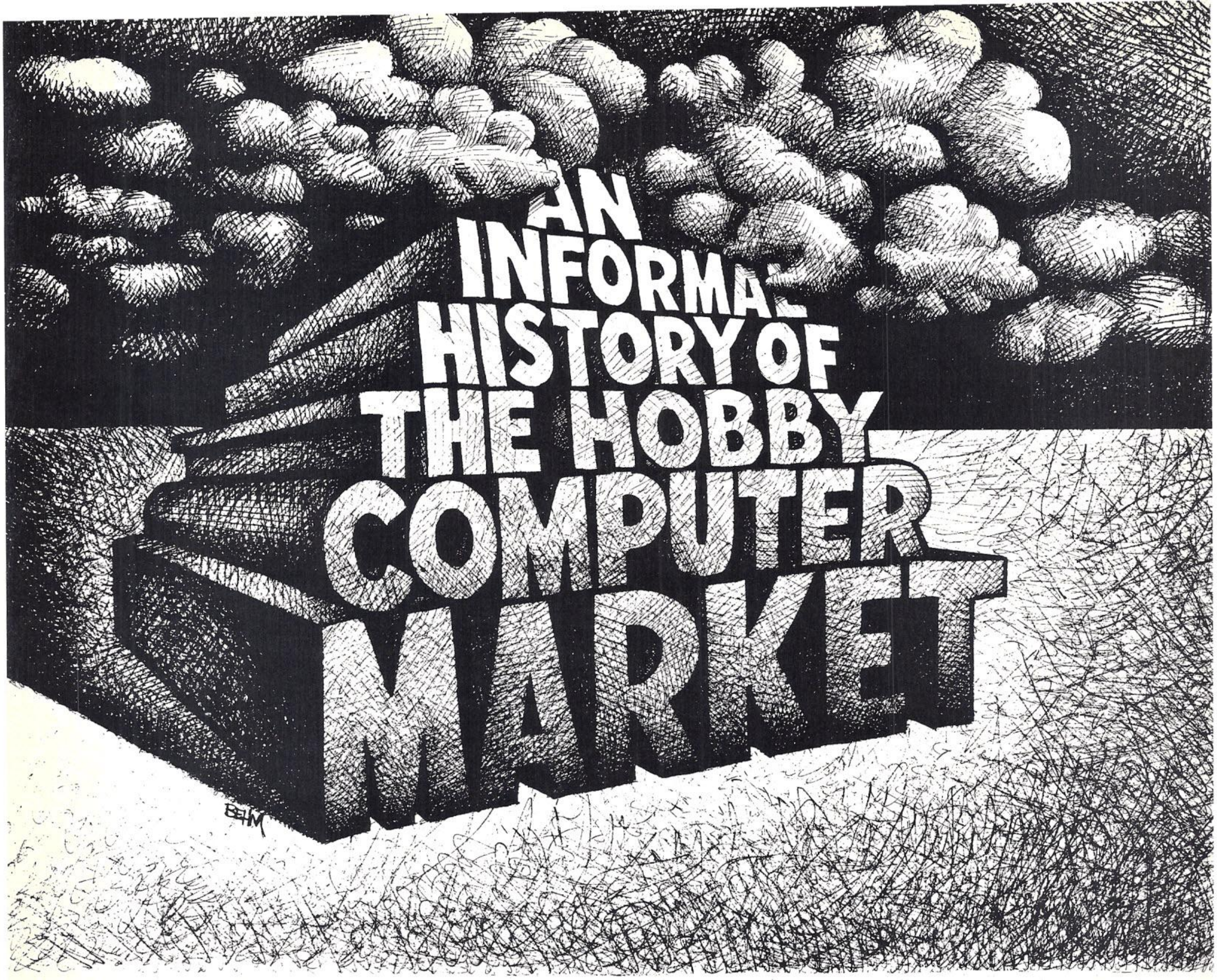
beginning. Then I have to use the string function to put "BRUCE" on the beginning of "LOMASKY". It would take four or five steps.

How do you feel about the thing you bought?

I'm very happy. The only thing I'm missing: I don't have the fun to play around. I can't pull out and put in boards the way you can with the Altair, the Imsai or the Sphere. But it has tremendous editing features you don't have with your Altair: I can scroll up or down and expand or contract a line. I can insert a word without having to retype the whole line.

Do you carry it around?

A lot. When I'm going to the shore for a weekend – my folks live at the shore – I'll throw it in the back of the car. It fits in the trunk. It's a little bigger than a large electric typewriter. An Altair with dual floppy disks and a videoterminal and a CPU would be three things to carry. Mine definitely is more portable, which is a big advantage, since I bring it everywhere I go. My wife complains I bring it in the bedroom too often.



by Alan R. Kaplan
Venture Development Corporation

This article expands upon the "History" section of The Home Computer *market study published by Venture Development Corp., Wellesley Hills, Mass. The author thanks the hundreds of computer hobbyists who were interviewed and who completed the company's detailed questionnaire.*

The hobby computer market was born in January 1975 when *Popular Electronics* ran a cover story on the MITS Altair 8800. Here's the interesting story of why that was the turning point and how the market has developed since then.

Origins (1963-1970)

Potential home experimenters (hobbyists) received their first tantalizing exposure to computers in the Sixties. The two most significant events during this period were (1) colleges installed timesharing terminals and instituted courses in BASIC and FORTRAN programming and (2) Texas Instruments Inc. introduced a series of integrated logic circuits (7400 TTL series) priced within range of electronics experimenters. In turn, two distinct interest groups took shape: one primarily software-oriented and another hardware-oriented.

These two groups proceeded on independent paths and rarely communicated with each other. Nonetheless, by the late Sixties both were quite large and highly motivated. Computer clubs existed even at the high school level (mostly to share BASIC programs), and amateur electronics publications invariably contained 7400-series circuits for simple counting and control applications. In 1967, Steven Gray of Darien, Conn. formed an Amateur Computer Society. By then a few hardy souls had succeeded in constructing rudimentary TTL computers with surplus core planes or "Rube Goldberg" mechanical memories.

Still, no hobby computer market developed, mainly because system-level hardware costs were beyond reach of the amateur, though they dropped rapidly throughout the Sixties. For example, Digital Equipment Corp.'s first minicomputer, the PDP-5, was introduced in 1963 for $30,000. In 1965, DEC introduced its popular 12-bit machine, the PDP-8, for what was then considered the incredibly low price of $18,000. By 1970, a basic PDP-8 configuration with 4K words of core was base-priced at about $10,000, but this price was above the amateur range.

Furthermore, few hardware- or software-oriented people would have known what to do with a mini even

if the price *had* been within range. Most software "types" were familiar only with interpretive languages, and even those with some assembly-level knowledge rarely knew anything about interfacing. Hardware types, slightly better off, had little or no experience with system-level software; they were generally at the digital logic level.

The Pre-Market Period (1971-1973)

The disciplines and interests needed to create a base for the hobby market began to consolidate. The jelling point came in 1971, when Intel Corp. introduced the first microprocessor, the 4004.

Although the 4004's short word size (4 bits or one "nibble") was adequate for electromechanical control routines, it was too slow for most other applications. The 4004 was also expensive, if indeed the hobbyist could obtain single units at all. But the message was clear: Intel had packaged a complete processor in one integrated circuit.

In 1972, Intel followed the 4004 with the 8008, an 8-bit (one-"byte") parallel processor. The 8008, with its ability to address 16K bytes directly, was even more significant. Though constrained by its 18-pin configuration and 8-bit data bus, it could take on many of the low-level tasks previously delegated to minicomputers in the way the 4004 could replace discrete logic. However, there were problems: it was expensive; its availability was uncertain; there was a severe shortage of know-how; and a support IC, the 8229, was delayed in production.

The following year, 1973, Intel introduced the 8080. It was 10 to 100 times as fast as the 8008, more flexibly packaged (40-pin DIP), architecturally advanced (with internal address register, 16-bit program counter and stack register), addressed more memory directly (65K bytes), and possessed a larger instruction set (74 basic instructions vs. 48 for the 8008). But 8080's didn't become generally available until the following year at prices in the hundreds of dollars. 8008's, however, were beginning to dip below the $100 level.

"101 BASIC Computer Games" outsold all other DEC publications. DEC still wasn't convinced that a popular computer market was imminent.

Popular Software (1973-1974)

The potential was clearly building for a hobbyist market among both the software-oriented and the hardware-oriented factions. On the software side, 1973 was a vintage year for the BASIC programming language. BASIC (for Beginner's All-purpose Symbolic Instruction Code) had been developed in the Sixties by John Kemeny and Thomas Kurtz at Dartmouth College, and for several years had been the preeminent language at schools. Almost every time-shared educational system contained a library of BASIC games and recreational graphics programs. In 1973, thanks largely to the efforts of Robert Albrecht, a West Coast computer "populist," and David Ahl, then at Digital Equipment, BASIC became the chief reason for non-students to want their own computers as well.

Albrecht, a "Johnny Appleseed" of the computer age, had acquired a PDP-8 and used it to introduce as many people as he could to computing. He installed his system in an artist's cooperative in Menlo Park, Calif., produced an inexpensive (65¢, now $2) text, *My Computer Likes Me When I Speak In BASIC,* and gave informal classes in BASIC to anyone, of any age, who would listen. Then he sent them out to teach others.

Albrecht's purpose was to "bring computers to the people" by first removing their mystique and then "turning people on" to the fun they could provide. Today thousands of computer hobbyists and even computer professionals owe their first lasting association with computers to the efforts of Bob Albrecht and his non-profit "People's Computer Company." In February 1973, Albrecht published the first issue of *People's Computer Co.,* a monthly tabloid of BASIC computer games, happenings and educational articles.

David Ahl deserves similar credit for cultivating a home computer market and, like Albrecht, used BASIC as his "seed". Ahl was Digital Equipment Corp.'s manager for educational systems and editor of *EDU,* a DEC professional publication for teachers. In October 1973, Ahl got his employer to publish *101 BASIC Computer Games.* This book was the first comprehensive collection of *tested* BASIC program listings for recreational use, and the games it included spanned a wide range of machine memory requirements and personal interest levels. Much to Ahl's delight and DEC's surprise, it outsold all other DEC publications within a year, and DEC found itself swamped not only with book orders but also with requests for quotations on minimum DEC configurations that would run the games.

Unable to convince management that a worthwhile popular computer market was imminent, Ahl left DEC for AT&T in 1974. In November of that year he published the first issue of *Creative Computing,* a magazine for computer users, teachers and observers of the computer's effect on society.

In 1974 three low-cost 8008-based microcomputer kits were available. None led to a popular market.

Popular Hardware (1974)

Some argue that the hobbyist market began in 1974 when at least three low-cost, 8008-based microcomputer kits were available. Unfortunately, none led to a popular market. One of them, however, the "Mark-8", came close. The Mark-8 was as significant a factor in preparing the hardware base for a popular microcomputer market as David Ahl's games book and Bob Albrecht's People's Computer Co. were in exposing prospective hobbyists to software and recreational applications.

The Mark-8 was designed by Jonathan Titus, a graduate student in chemistry at Virginia Polytechnic Institute. After experimenting with the Intel 4004 in 1972, Jon in 1973 designed a hardwired "home brew" system around the 8008. Later that year he sent an article on his system to *Radio-Electronics* magazine. Appearing in the July 1974 issue, the article was accompanied by an offer to provide a printed circuit board and a book of experiments. Over 1250 readers wrote to request the PC board (at approximately $50) and at least three times as many purchased the book for $5.

Independently, David Larsen and Peter Rony, who were teaching chemistry at VPI, had prepared an excellent series of "learn-by-doing" experiments in digital logic. Their *Bugbook I* was published in August 1974, after a tie-in was arranged with E&L Instruments,

expensive assembled units. Roberts had Inc., Derby, Conn., to package the necessary hardware. Titus, who had set up Titus Labs and was doing government work on microcomputer interfacing, informally joined with Larsen and Rony to design the hardware. Titus then produced *Bugbook III,* a similar learn-by-doing collection of interfacing experiments based on an 8080 system to be called the "Mark-80." E&L Instruments packaged the Mark-80 and re-named it the "Micro-Designer."

Bugbook III and the Micro-Designer were not available until 1975, but before the end of 1974 over a thousand Mark-8's were in the hands of home experimenters. Titus and the authors of *Bugbooks* I and II had provided the

MITS hoped to sell 200-300 units in 1975. That projection was too low by more than an order of magnitude.

hardware and documentation for an intelligent but electronically ignorant hobbyist to construct his own system, one on which he could learn digital logic and machine-language programming. Before Mark-8, any market for basic design kits would have been only an extension of the electronics hobby market. Unless "already into" electronics, a hobbyist was unlikely to "get into" *building* an electronic computer. Mark-8 cleared a path for everyone.

By the end of 1974, the hobby market was a coiled spring waiting to be released. Exciting new microprocessors, notably Motorola Corp's M6800 and National Semiconductor's 16-bit PACE, were described in trade (and financial) publications as "opening the door to a computer in every home." BASIC interpreters were being implemented on microcomputers. A growing number of working applications, in contrast to systems that merely worked, sparked the interest of the non-computer oriented. A demand arose for low-cost terminals (particularly the Teletype ASR-33) and for software. Commercial minicomputer suppliers noted that demand for products for personal use increased in both quantity and quality.

Purchases of minicomputers for personal use were still rare enough to rate press releases ("Digital Equipment Helps Boy to Acquire Own Computer"; "Glendale Man Uses Computer to Pick Horses"). But now individuals were requesting software and low-speed peripherals, often for the stated purpose of interfacing to microcomputers.

All this activity still didn't add up to a commercially important popular market. Notwithstanding the yeomanry of Titus, Albrecht, *et Ahl*, there were still too few hobbyists with the right combination of means and inclination for any established manufacturer to risk developing a low-cost, general-purpose microcomputer system for popular use. By every common sense dictate, such an effort *would* have been premature. No one by this time doubted that a hobby market would develop; the question was when it would materialize and whether a "volkskomputer" would generate its *own* market or would appeal only to the highly motivated few who were interested already.

In 1974, the established computer industry was having trouble enough meeting delivery dates for non-hobby applications. To release the coiled spring required an established company small enough to commit itself to developing a new market yet with sufficient means and experience to do so intelligently.

The Market Emerges (1974-1975)

Micro Instrumentation and Telemetry Systems, later known as MITS, had been founded late in 1969 by H. Edward Roberts. It had developed a few products for radio telemetry, then produced what is generally believed to be the first calculator offered in kit form to the consumer market.

When the calculator market sagged, Roberts began looking for another product. The success of Titus's Mark-8 *via* the *Radio-Electronics* article was all he needed to put into motion his own idea for an inexpensive computer in kit form.

Roberts rushed to complete the design of what was to become the Altair 8800 by Christmas of 1974 and arranged with *Popular Electronics* to do a cover article. The article didn't appear until January 1975, which disappointed Roberts, since he had been depending on Christmas mail orders for a large part of the 200-300 units he hoped to sell in 1975. As it turned out, that projection was too low by more than an order of magnitude. MITS' 20 employees were soon deluged with orders for its \$395 (later \$439) kits and more greatly underestimated the market for a low-cost ready-to-run microcomputer.

In the next few months, Roberts more than doubled his staff and blanketed the computer trade and electronics hobby publications with ads for the Altair. Roberts was aware that potential competitors had noticed its popularity, and he was determined to pre-empt the fledgling market.

To accomplish this objective, he first committed MITS totally to penetrating the hobby market at *all* levels. This meant developing a BASIC interpreter for the Altair that would attract those who wanted a system they could interact with immediately. Second, he challenged the commercial sector even before existing microcomputer manufacturers could react at the hobby level. A full-page advertisement in the April 1975 issue of *Digital Design* magazine, for example, offered an "Advanced Accounting/Engineering System" consisting of "an Altair with 32K of memory, serial interface, Teletype (or terminal), line printer, disk controller and disk drives, DOS and extended BASIC software" for \$10,489.

Whether MITS could deliver and support these "Advanced Accounting/Engineering Systems" is a matter of conjecture, but Roberts' reading of the market for a ready-to-run BASIC system was on the mark. Altair offered something for every level of hobbyist. By the summer of 1975, orders from hobbyists were arriving at a rate that, according to MITS then-VP of Advertising, David Bunnell, on one Friday afternoon exceeded 300.

A Clear and Present Market (1975-1976)

What had been an "emerging market" in the first half of 1975 became an "emerged market" in the second half. That summer the Computer Store Inc.,

IMSAI was one of MITS' most unrelenting rivals. IMSAI called itself "rugged."

a MITS retail outlet, opened in Santa Monica and was soon followed by several non-affiliated retail stores.

Carl Helmers, a knowledgable and innovative hobbyist who had been publishing a newsletter, and Wayne Green, publisher of an amateur radio magazine called *73*, launched a new "slick" magazine, *Byte*, for amateur

computer enthusiasts.

The increasing popularity of programmable pocket calculators whetted appetites for more programming power. The IBM 5100 portable BASIC/APL computer, introduced in September for under $9000, further educated the public to the viability of the "home computer" concept.

A new organization, the Southern California Computer Society, grew almost overnight to encompass a national membership in the thousands.

In the second half of 1975, MITS faced competition from firms that were able to capitalize on and improve upon the weaker features of the design with which MITS had hurried to market. Notable competitors were IMS Associates Inc. ("IMSAI"), Southwest Technical Products Corp., Processor Technology Corp., Sphere Corp. and the Digital Group, Inc. Cost and product quality became competitive points as the new systems were introduced, and MITS began to produce improved second-generation equipment in a maturing market.

One of the most unrelenting of these new rivals was (and is) IMS Associates Inc., whose IMSAI 8080 advertisements were replete with such words and phrases as "rugged" (three times in one full-page ad), "durable", "heavy-gauge" and "commercial grade". IMSAI was priced $160 above the Altair 8800, but this was more than made up in the view of many hobbyists by the IMSAI's design

Computers will become almost as ubiquitous as the CB radio. The marketing question is no longer "should" but "how."

improvements and much heftier power supply. However, the IMSAI 8080, like several others of the newer systems, was designed to be bus- and PC card-compatible with the Altair 8800: a memory or interface board for the IMSAI 8080 also works in the Altair 8800.

The cross-compatibility amounted to nothing less than a *de facto* standard, opening the door to any number of smaller firms capable of producing add-in memory, directly compatible peripherals and even alternate processor boards. In less than one year, the hobby computer market had evolved to the point of supporting a PCM (plug-compatible manufacturer) infrastructure at the board level.

This development is as important as the original Altair announcement because it indicates the pervasiveness and staying power of this young industry. Indeed, the impetus it provides will extend far beyond the hobby market, and it will only be a matter of time before the machines that barely a year ago were considered novelties will be almost as ubiquitous as the CB radio, at least as much a fixture in the home as the microwave oven and as common in the small office as the editing typewriter. Their appeal is not merely that they are affordable but that they are *understandable.* They can be used by preschooler as well as professional and are limited in application only by the imagination.

Today's hobby computer is nothing less than the leading edge of the consumer revolution. It cannot be ignored by any company presently serving the computer industry. The marketing question is no longer "should" but "how".

MAGIC AS THE MIDAS TOUCH

–Henry Gilroy

You need only say a cheerful word about sunny weather to bring on a rainstorm or complain about human meanness to receive a gratuitous kindness from a stranger. It's a perverse world.

In The Equalizer, a column on another page of this issue of PERSONAL COMPUTING, *a wry complaint is offered that the computer graphics field is at a frustrating standstill, that artists are adapting themselves of necessity to the limitations of computer systems and pretending that the styles forced on them are art.*

No sooner had these wicked thoughts flowed through the typewriter onto the paper than there appeared in the mailbox a clipping bearing an advertisement for General Electric Co.'s Genigraphics. The services offered by this company substantially refute the Equalizer complaint.

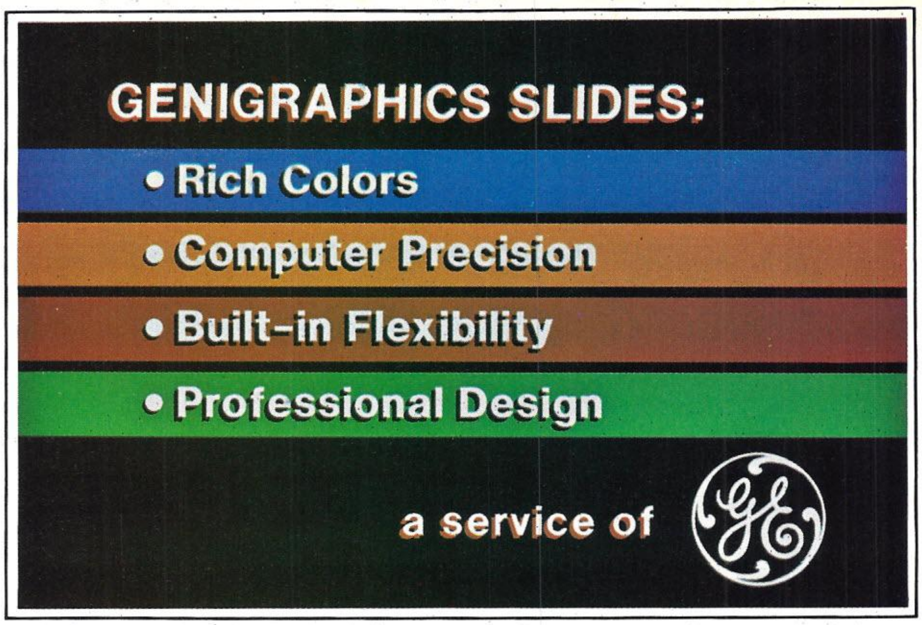
This adaptation of the computer to practical labors is "personal" in the sense that any customer may walk in off the street with a modest amount of cash, cause the computer system to perform to his specifications, and carry away the product of that performance.

That's "personal computing" well worth earnest attention.

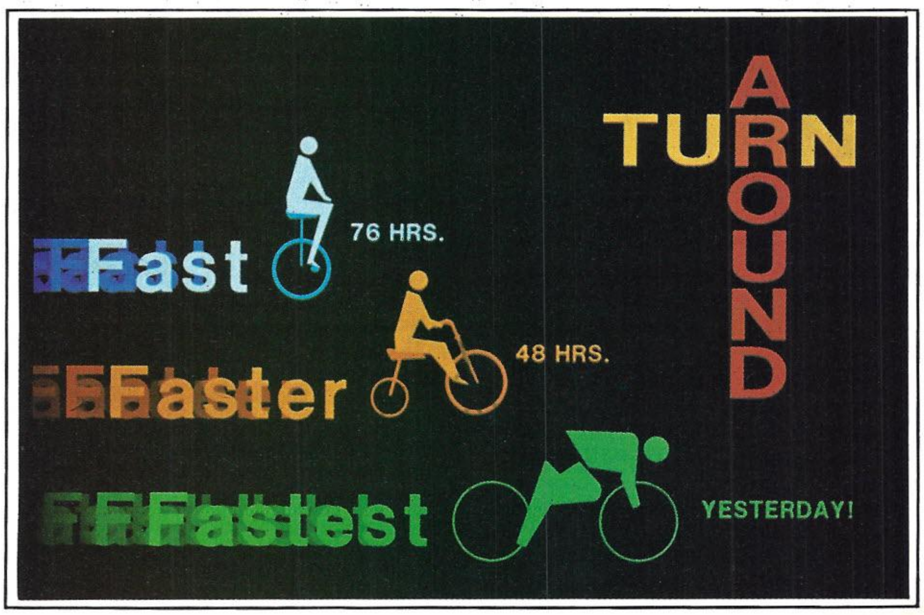
For centuries, people have employed complicated machines as drawing instruments to do the repetitious labor that is necessary in laying out formal plans, charts, graphs, lettering and the ordinary display materials of the workaday world of commercial art.

In this century of sales presentations and stockholder meetings, when standards of graphic excellence are influenced by constant exposure to television, magazines and movies, the pressure to produce well-designed and handsomely rendered work is very great. Cost and speed are

This Genigraphics promotional slide looks like the product of 15 minutes of layout, a trip downtown to the title house and another to pick up the finished cels with opaque lettering in black and white, half an hour of cut-out and pasteup, a trip to a photographer for shooting the slide (probably a two-day turnaround like that at the title house) and a final trip to the lab to pick up the mounted slide. But in fact, the Genigraphics operator can knock out the completed artwork – with titles, drop-shadows, logo, background colors and all – in a few minutes of work at the console.*



The figures on the bikes are variations on a single basic figure stored in memory. The operator can manipulate such figures – distort them, change colors and build new artwork – right at the console. In reality, of course, he is manipulating digital data with instructions to the CRT on what to display. Any false starts (discards) waste time and energy, not substances.



the controlling factors in this game.

As soon as digital computers became available in the late forties, the people who had been struggling to use analog systems, Lissajous figures and the like realized that the new machines might serve their needs.

They were not simply trying to set type automatically. They were trying to equip artists with fast, responsive systems that would speed the trial-and-error process that is at the heart of commercial art.

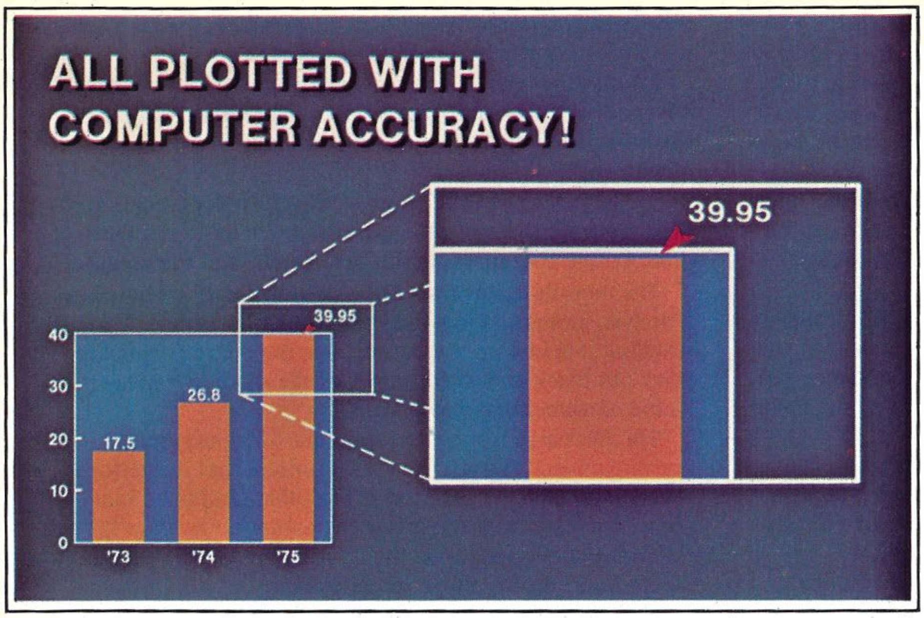
Typically, the artist prepares material for a client who can't do it himself, either because he hasn't the skill or time. The client's driving concern is that he needs a set of slides for a meeting, and he won't have the information that must be on those slides until it is almost too late to make them. He also knows that he wants the pictures to look "good, professional," as well as conveying information efficiently. He needs color, design, some flair. He has a small budget, and all he can do to help the artist is wave his arms and make ambiguous sounds.

Usually, the artist must sketch the material out, sometimes in almost-finished form, and show it to the nervous client before they can communicate. Trial and error.

Ideally, the mechanical aid to the artist will let him draw *finished* work, complete with color, type and symbols, in just a couple of minutes so the client can make necessary changes, wait another couple of minutes and approve the finished work on the spot. Traditionally, the artist has to start over after changes to do a piece of finished work *based on* a semi-approved rough. He'd prefer to use what he has already done instead of starting from scratch.

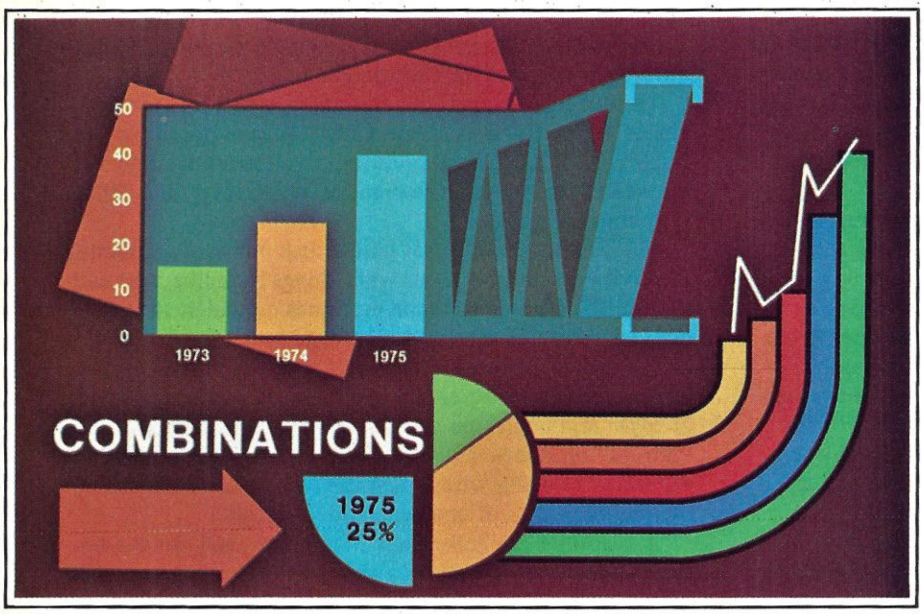
Obviously, the computer permits this to be done, doesn't it? The computer operator need only call type from memory, specify its size, color and location, and display it on a television monitor. Simple. Backgrounds can also be called from memory and slid in behind the type, right? Standard symbols like stars, human forms and buildings can be spotted wherever the operator wants them.

The client can see what it looks like, call for changes that



*Even an artist with a protractor finds pie charts a tiresome chore and strikes compromises that are accepted reluctantly by nervous clients. "Divide these lines each into 17 equal parts." "OK, *sigh*."*

The computer handles these tasks automatically, splitting hairs finer than the client's eye can perceive. A dozen highly accurate trial renderings are no trouble. The client can see from the preliminary work exactly "how it will really look."



The client at a distance can handle this production work by mail, if necessary, with some hope of getting finished work that is as good as he hopes.

All of the standard forms, figures, typestyles and colors can be chosen from a catalog and specified on a simple layout sheet. The operator trims up the designs, combining and fitting all the elements to a human eye.

Even the long-distance customer has access to sharp, fresh professional design work . . . if GE continues to provide useful services. If not, others will, of course. This is a pioneering system.

the operator instantly makes, approve the composition, shoot a photograph of it and go on to the next. The concept is simplicity itself.

But questions arise.

- How much would it cost?
- Do you teach an artist to operate the system or teach a computer operator to be an artist?
- How flexible is the system? How many type styles can you choose from memory, and how much can you manipulate them? What is your range of colors? Is the red color you're using now going to match the red color you got from the system last time? How do you draw non-standard symbols? Can you insert a photograph of the company president for the background?
- How much must you tell the machine? Suppose you want to make a pie graph in which the wedges add up to 100% of the circle. Must you tell the machine exactly how long each line is, or can you just ask for 12 1/3% of the circle and let the machine calculate and display it?
- When you move some elements of the image around, is it hidden behind other elements in some places, or does it cover them . . . or do you see both images as semi-transparent ghosts?

And so on. The practical concerns in creating and using a drawing machine of this kind are staggering. They'd be reduced somewhat if everything could be limited to simple, standard layouts and the client could be satisfied with conventional forms. Unpredictable aesthetic judgements are involved, however, not just math, physics and tradition. No capability is ever quite enough. The needs and opportunities are endless.

The chaps at General Electric Co. in Syracuse have labored over these problems for a couple of decades. Their Genigraphics package of hardware and software is not only a portent of the future, but a practical system that produces useful things now. The future is here.

GE's installation in Hollywood, Calif., where demand for commercial graphics tends to be high, is in the building com-

plex of a major film-processing laboratory. The visitor parks his car somewhere within a radius of 10 miles, picks his way through a maze in the laboratories and climbs a long, steep iron stairway to the Genigraphics center. The laboratory always smells slightly of warm developer, whose sensory input adds interest to the visit.

The small establishment is crowded and busy. Ray Christopher, the aggressively enthusiastic manager of the center, is about to demonstrate the system for potential clients. The lucky visitor can sit in.

Christopher insists that he's a poor operator, slow and out of practice, but his hands move fast and surely over the controls on the operating console, typing in commands and data, moving the joysticks to position elements of the image where he wants them.

The image is formed on a color television monitor centered in the console above the control board. The monitor is perhaps 12 inches in diameter, a comfortable size for display of the work in process.

Christopher insists that he's slow and out of practice, but his hands move fast and surely over the controls on the operating console, typing in commands and data, moving the joysticks to elements of the image where he wants them. it. Whatever the cursor touches, he can control. The cursor alights on one of the flat faces of the object and Christopher touches a couple of keys. That surface on which the cursor was resting is no longer plain white. It's colored. He twists a knob and the tone of the color changes.

Rapidly, he moves the cursor from surface to surface, fingering the controls lightly, coloring and shading each area. His Midas Touch can turn any facet of the object to gold . . . blue . . . red . . . or any of thousands of color mixes that suit his eye. (An alphanumeric display on the console reads out codes representing color and tone so that colors can be matched later by input of the same codes.) When he is satisfied with the object, Christopher moves the cursor to the background and zaps in a color. As by magic, the object called from memory has assumed a three-dimensional quality and is floating gorgeously in a colored sky.

Christopher wipes away his creation and repaints it so it is altogether different. He replicates the image, expands it, contracts it, distorts it, places it where he chooses, strips in type from memory, fashions words, columns of numbers, special effects. He mingles images, causes them to disappear and reappear in metamorphosis.

He calls a rectangle, the frame of a graph. "What scales do you want in X and Y?"

"Tick marks or full lines?"

"What values?"

"Bar graph? Flag chart?"

"Positive values in one color, negative in another?"

"Cumulative?"

"Divide the line into 17 equal parts?"

"Log scale?"

"Add symbols . . . zoom in on one section . . . add arrows . . . perspective lines . . . upper case . . . lower case . . . all letters in different colors . . . repeat a phrase . . . "

Zip, zap, let it be done.

Alter, adjust, increase, destroy, recall.

Ray Christopher works like an artist with a fine instrument in his hands. Indeed, he *is* an artist. So are all of the operators. They're not computer experts graciously bowing to the unreasonable demands of their customers but experienced artists working with their clients to create whatever is desired. When pressed with technical questions, the operators shrink back and beg off. The real technical people, the engineers and programmers, are back in Syracuse, working away steadily at improvements to the system.

Dick Lindeman is one technical man in Hollywood who hovers over the system attentively, maintaining it, looking for flaws, keeping it at work. He's the man to ask about technology.

Lindeman is proud of the system and offers a quick technical tour. He describes the system's three basic elements: the console, the computer and its peripherals, and the recorder.

The console offers a keyboard with an array of buttons, switches, knobs and joysticks with an LED display. A copy machine tied in with the console lets the operator quickly print out black and white copies of the monitor's image, so the customer may have proofs.

The television display is a standard 525-line system, nothing special, though the visitor notices that image focus is very good all over the face of the tube. Lindeman is at pains to explain that the image the computer holds is far better than the one visible on the television monitor, which would never do for photo reproduction.

A blown-up picture of that tube would clearly show the 525 lines, and the resolution of the image would be very unsatisfactory. The image on the console monitor is just a working rough, not the final product.

The console controls the computer, a PDP 11/35 with 32k of 16-bit words in RAM. Only 24k is required for Genigraphics handling of still pictures, says Lindeman, and 28k is required for movie animation, of which they do a deliberately limited amount.

The computer converses with a disk memory containing all it has been taught about what things look like, and with a couple of DEC digital tape machines on which all the image data can be recorded and read.

An odd thing about this system, from the artist's point of view, is that he never has any flatwork to store, no pile of paper and board, no cel. The system can store up to 56 "images" on a little four-inch DECtape reel. (The images are stored as codes with which the computer can instruct the recorder, not, of course, as video signals.)

A customer can come back in a month, call out the images from his file stored on a reel, alter them as he chooses (changing dates, values, names, whatever), take away fresh slides of the updated work and leave the altered image stored on the reel.

A teletypewriter and a line printer are on line for conventional input and output. Logs of the stored material, color values and similar information can be printed out on paper for easy reference.

No, they don't scan existing photographs or art to read them in. Creation of a new typeface or symbol in memory is a big chore.

Off in a dark room is a Docomed recorder with appropriate interfacing that transfers the images to film. There are two standard cameras on the recorder, one that shoots motion picture images on 35mm film, and another that shoots 35mm slides. (The worksheets in the handbook also refer to superslides and other formats.)

Note that movie frames are arrayed vertically along the film, each four sprocket holes high. They are much smaller than standard 35mm slides, which are arrayed horizontally on the film, each eight sprocket holes wide. The height-to-width ratio of slides and movie frames is slightly different, giving the pictures a slightly different shape. This difference is just one more curse to which the system and the operators must be adapted.

The recorder uses a five-inch cathode ray tube with a

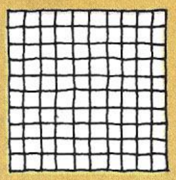
Computer graphics tough for amatuers

Personal computing systems are still extremely limited for creating useful pictures. Neither dealers nor manufacturers have been willing to recomment an experimental package of graphics equipment and software for the amateur.

The amateur has one advantage over commercial developers: he's not necessarily in a hurry. If his system spends all night processing a single picture, he may be satisfied. Nobody can afford that commercially.

Consider some of the basic realities of graphics before tackling the picture business.

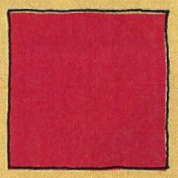
For example, what *is* a picture? The case of black and white pictures printed in a magazine like this one is illustrative (no pun intended). Notice that we have black ink and white paper, not various shades of grey ink on white paper. These letters are all uniformly black. Printed line drawings are made of lines that are uniformly black. Photographs are printed with the same black ink, but you have the illusion of "shades of grey" because of the "half-tone" technique developed for the purpose. It's a matter of statistics.



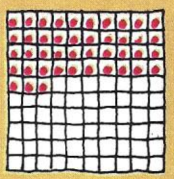
Set up a grid. This one is ten by ten.

Assume that the lines are not visible. This field appears entirely white to the eye. No ink.

Now switch. Fill in each of the squares with ink. The field is completely black. All ink.



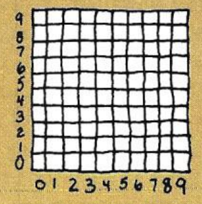
Now our dynamic range has been determined, the limits of light and dark in our photograph. Everything else must be something in between. Now, in each square, draw a dot that fills exactly half the area of the square.



If you get far enough away from this drawing so that you can't distinguish the separate dots, the whole field will appear grey. You're still using just white paper and black ink, but by controlling the size of the dots you can determine the shade of grey that the viewer perceives. If you break your photograph into very large numbers of dots and control the dot size, you can print grey-scaled pictures with black ink on white paper. Perfectly straightforward. Color pictures are printed by laying on one color of ink after another. The main problem is to print the successive color images in the right locations.

Look back at the grids you drew and notice that the dots are all evenly spaced, center-to-center. That's true of printing halftones, too. The size of the dot is the variable. That means that you can assign x and y coordinates to the dots to identify their locations easily. In this ten by ten grid, very crude, we have only a hundred dots.

Old-fashioned, cheap newspaper cuts (halftones) were usually made at 60 lines to the inch. A picture filling a square 2½ inches on a side contained 22,500 dots, and the picture quality was nothing to rave about. Highgrade publications like the *National Geographic* magazine run pictures with resolution between 125 and 160 lines to the inch. A single square inch at 150 lines contains that same 22,500 dots we talked about in the 2½" by 2½" square above.



That's a lot of dots.

If you want to think in terms of dissecting an image (or synthesizing one) and storing it in the computer memory in the form of bits, you'll need a lot of memory. You'll need information on the location of each dot and its presence or absence (assuming that each square is either filled with ink by overlapping dots in adjacent squares or left entirely without ink).

If you want to deal with grey scale, you'll have to add information on the size of the dots. To achieve 64 levels of grey within the dynamic range established between white and black, you'll need to store six bits plus the location information.

And then, there's color.

Clearly, even the system discussed in the Midas article doesn't deal with pictures in this way. That's part of the reason they don't have a scanning system that will read photographs into the memory.

Just to get a feel for this, consider the terrible image presented to you by your home television receiver in black and white. Though the station broadcasts images that are 525 by about 600 lines in resolution, the typical home receiver a few years old shows a picture about 300 by 400 lines.

If you stored just one of those in straight, high-contrast binary form, you'd be storing 120,000 bits. If you use a six-bit grey scale, you'll be storing 720,000 bits. If you want to process such a picture in your computer, you'll be able to take the thing line-by-line, perhaps, and spend as long as necessary doing something systematic to the picture to alter it . . . or to pick information from it if that's what you want.

Still, the amount of data you must process is overwhelming, and we haven't addressed the problems of programming, hidden lines, color and so on.

Computer graphics is a tough field. The amateur should know what he's getting when he hooks up even crude input/output devices like Cromemco's Cyclops camera (with a 32 x 32 field) or Dazzler television display (with 128 x 128 color field). It will be a long, hard road even to crudely manipulated crude pictures.

But we know that fine things can be done and they will be done ever more easily as hardware and software develop over the years. The field is wide open to amateurs with patience.

Lots of patience.

white phosphor. The resolution of the CRT is 4000 lines, compared with the 525 lines on the monitor, and the images recorded from it are about as good as the film can resolve. No problem with large projection.

Color is produced with a "color wheel," a sectored disk spinning in front of the CRT at 1800 rpm. The sequence of primary color filters is repeated on the disk so that each primary color passes by 3600 times a minute. Timing is a delicate matter but well within electronic control capability.

The flying spot traverses the CRT, controlled by the computer. (The computer can think about only one job at a time, says Lindeman. You can work at the console or with the recorder, but not simultaneously.) For each element of the picture it is recording, the spot flashes on and off to expose the appropriate amount of red, yellow and blue to produce the desired color and intensity of color.

The process of exposing an image on film varies in length, depending on the amount of physical detail and color complexity. A minute is about par for the course, but Lindeman was impressed that a recently run slide had required 4½ minutes in the recorder. One fabled slide is supposed to have consumed 15 minutes all by itself. A legend, maybe. But compared with hours and hours of human labor over a drawing board? Not bad.

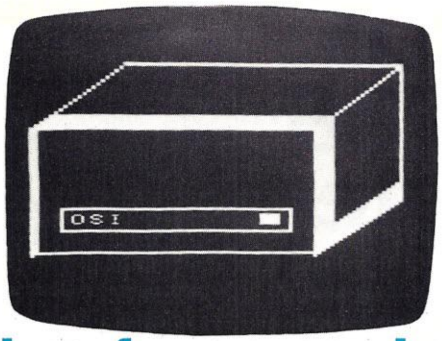
General Electric not only offers Genigraphics service at three centers around the country but sells full systems to those who feel they have the volume to justify the "third of a million" cost. (If you operate in a city that has no lab that can handle the film processing, explains Lindeman, you may as well save a hundred thousand or so by not buying the recorder and sending your data by wire instead to New York, Syracuse or Hollywood for recording there.)

The company publishes a newsletter, *Genigraphics User,* that is circulated to customers. Comments in the letter are revealing. While pointing to improvements, the customers write about matters that must have been sore points until the change. For example, from a December 1975 issue: "The minor distortion difficulties encountered with PTP line have been eliminated" or "Capturing one of a number of tightly clustered vertices is now much easier." Try: "The cursor response-to-joystick motion has been improved."

Genigraphics is hardly hobbyist activity. The resources of a big, strong company were required over a period of many years to develop this kind of system and put it to work in the field. The average price of a slide, delivered to a customer in its plastic mount, is around $30. Turnaround time is typically three days, whether the customer is buying 3 pieces of work or 50. That price is highly competitive, and not many commercial artists can promise 50 items in three days with the same confidence they promise three. This computer system is very different, a really new approach to classic problems.

Better than the Midas Touch.

Meet the Challenger.™



The Challenger Self Portrait

The new price and performance champ from OSI.

He's got his act together!

Even our lowest-cost Challenger comes fully assembled, complete with a 500 ns 6502A, serial interface, 1,024 words of memory and a UL-approved power supply, all for $439. Every Challenger comes ready for easy expansion with an 8-slot mother board, backplane expansion capability, and a power supply heavy enough to handle a full complement of system boards. Our 4K Challenger comes ready to run BASIC minutes after you unpack it. And there's more.

He packs some heavy hardware.

You've never seen memory and interface options like these—not at our prices, fully assembled! 4K RAM memory boards $139! (see below). Single drive OSI Challenger Floppy Disk $990! Dual drive Floppy $1490! Plus 8K PROM boards! A Video Graphics board, including alphabetics, graphics, and **color!** An audio cassette, A/D, D/A and parallel I/O board! A backplane extender board! A prototyping board! And our extraordinary **CPU Expander Board**—it lets you run a Z-80, and 6100 (PDP-8 equivalent) **concurrently** with The Challenger's 6502, or under its control.

There's nothing soft about his software!

OSI has full software support for our Challengers. Including extended BASIC, extended Video Monitor, a Disk Operating System, some very Hollywood real time programs for Video Graphics, Animation, Sound Processing and so forth, plus PROM firmware, with more to come.

He's fast!

You can order The Challenger with a 6502C for a 250 ns cycle time, with a standard 6502A for 500 ns cycle time, or with a 6800 for 1 microsecond cycle time. And with our CPU Expander Board, you can always update to any new CPU to be as fast as fast can be.

And he isn't just good!

He's better! By design. The OSI Challenger is the only completely-assembled, ultra-high-performance, fully-expandable mainframe computer that does this much for this little. Get your hands on one now. Send for your Challenger today.

You can't beat The Challenger!

The OSI Challenger 65-1K. Fully assembled. Features 6502A CPU, serial interface, 1,024 words of memory. **$439.**
The OSI Challenger 65-4K. Same as 65-1K but with 4,096 words of memory. Will run Tiny BASIC without expansion. **$529.**
The OSI Challenger 65V-4K. NO NEED for an expensive terminal. Connects to your ASCII keyboard and video monitor through included OSI 440 Video Board. Features software utility that simulates a deluxe CRT terminal. **$675.**
The OSI Challenger 68-1K. Based on 6800 CPU. For the casual hobbyist, smaller systems. The Challenger 68 series comes only in serial interface forms and is compatible with MIKBug software through an included OSI software utilities package. **$459.**
The OSI Challenger 68-4K. With OSI 4K BASIC on paper tape. **$529**
SPECIAL! ADDITIONAL 4K MEMORY BOARDS. Ordered with your Challenger, limit 3 more at this special Low Price, (total 16K, including 4K already on-board in mainframe). **$139**
Buy 12K or larger Challenger 65 system and we include Extended BASIC FREE!

OSI Challenger Floppy Disk System. Fully assembled, for use with OSI Computers only. **$990** Single drive **$1490** Dual drive.
OSI Audio Cassette Interface. Comes assembled, but with room for you to populate with A/D and D/A chips later. (OSI 430 based) **$89**
And all the baseboards and kits of the powerful OSI 400 System.

OK, OSI, I'm ready to buy!

To order your Challenger System, send the total amount of your purchase plus $4.00 for shipping and insurance (plus sales tax for Ohio orders) by personal money order or check. Or indicate **all** numbers on your BankAmericard or Master Charge to charge your order.
Or send a 20% (non-refundable) deposit to receive your order C.O.D. Delivery is typically 60 days (except when payment is by check, which must clear before shipment can be made). Deliveries are scheduled on a first ordered, first shipped basis.

Name________________________

Address______________________

City__________State_______Zip_______

Telephone____________________

Bank card info Inter Bank #___________

Expiration Date________________

Account # ____________________

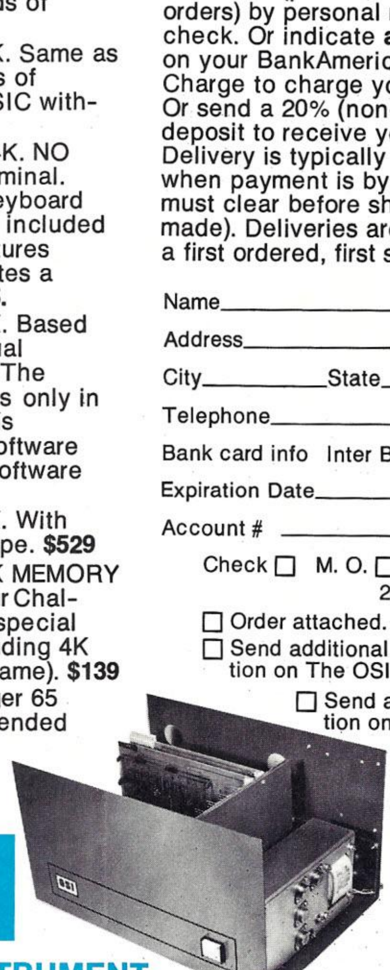
Check ☐ M. O. ☐ BAC ☐ MC ☐
20%, bal. C.O.D. ☐

☐ Order attached.

☐ Send additional information on The OSI Challenger.

☐ Send additional information on OSI 400 Kits.

☐ $1.00 enclosed for complete OSI Catalog.



OSI
OHIO SCIENTIFIC INSTRUMENT
Dept. PC 11679 HAYDEN STREET, HIRAM, OH 44234 (216) 569-7945

CIRCLE 23

The POLY 88 Microcomputer System

The POLY 88 is not for everybody; but if you are into computers, or considering getting a system, the POLY 88 is a machine to think about.

For the hardware buff:

• Popular 8080 central processor • Single-board CPU with vectored interrupt, real time clock, single step logic, and serial I/O • Video interface card to handle communications — generates video to TV monitor and provides parallel keyboard input port • Serial and cassette interfaces on small mini-cards that plug directly into CPU board with ribbon cables • 300 baud cassette • ROM monitor with powerful debugger, video software, and bootstrap loader • Backplane and power supply on one board simplifies construction • Rugged 6 amp power supply • All circuit boards are high quality double sided with plated through holes • System is compatible with a wide range of Altair peripherals on the market • Minimum point to point wiring means that the POLY 88 kit can go together in three evenings!

Front Panel Display:

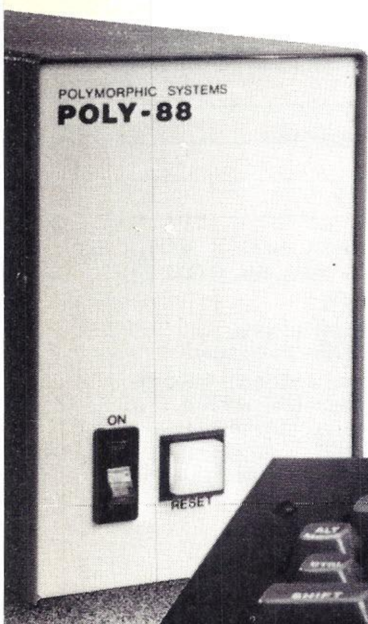
Routines in the 1024 byte monitor display the contents of each of the 8080 internal registers, and the value in memory that is addressed by each register pair. Programs may be executed one instruction at a time. Data at any location in memory can be displayed and may be easily altered. All front panel data is entered in hexidecimal notation for operator convenience.

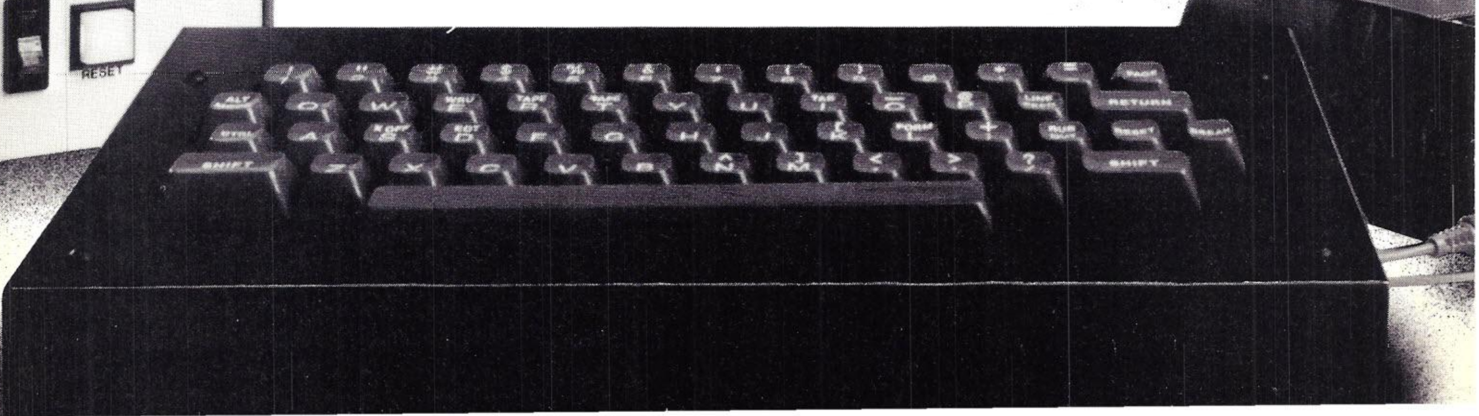
For the software buff:

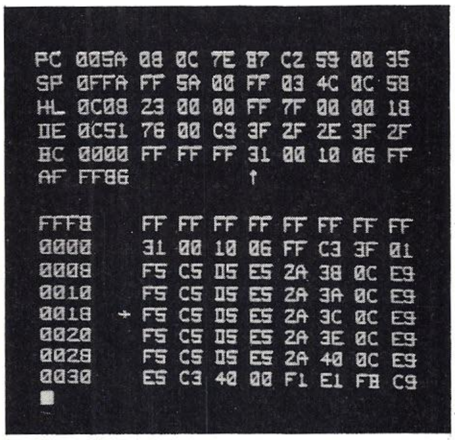
Software is what the POLY 88 was designed for. The user can go all the way from using higher level languages like BASIC to developing machine code with the aid of our assembler. Our BASIC is a full 8K BASIC with character and byte manipulation capabilities; and it is designed to run on our system. No kludging up or special fixes needed to run on multiple I/O devices. Best of all, the programmer is finally free of the teletype emulation mode so the video display can be used to full advantage. The video display provides a unique opportunity to write new types of programs and games. Characters (16 lines of 64) and graphics (48 by 128 grid) are part of the processor's memory, so the display may be altered extremely fast — less than 20 milliseconds to write the entire screen. The 1024 byte ROM monitor provides many I/O handling routines, leaving the programmer free to concentrate on his particular application. The POLY 88 hardware provides many of the additional features that programmers have come to expect from computer systems, such as vectored interrupt (which allows multiple concurrent I/O handling) and real time clock. These features are standard equipment and are included in every POLY 88. So, whether you want to develop a new computer language or fight Klingons, the POLY 88 hardware provides a firm foundation on which to realize your programming fantasies.

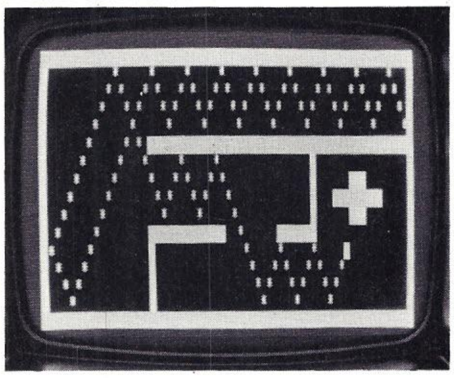
For the person new to computers:

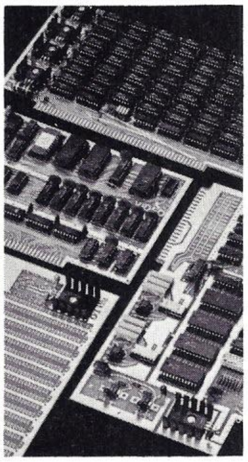
For those of you who are just getting started with computers the POLY 88 is a machine that was designed to be easy to use. No complicated hardware setup and software fixes needed. No one should have to learn all the inner workings of a computer just to enjoy one at a reasonable price. With the POLY 88, you can "play" pre-developed programs and then explore writing programs in either BASIC or assembly language, as your interests expand.











POLY 88 System Prices

System 1 kit includes 8080 vectoral interrupt processor board with real time clock, ½K of RAM, and 1K monitor on ROM; Video Terminal Interface for displaying 16 lines of 32 characters on video screen and inputing keyboard signals; cabinet, backplane, and power supply; complete assembly, theory, and operation manual. $595.
System 2 kit includes all items in System 1 and a Byte/biphase cassette interface kit. $690.
System 3 kit includes System 2 plus 8K of RAM with BASIC and assembler programs on cassette tape. $990.
System 4 is the complete kit. It includes System 3 with TV monitor, keyboard, and cassette recorder with all necessary cables and connectors. $1350.
System 7 is System 4 assembled, tested, and ready to run. $1750.

Accessories:

8K RAM kit. $300. Assembled $385.
POLY I/O Ideaboard, hardware prototyping kit board. $55.
Analog Interface (1 channel) kit. $145.

All prices and specifications subject to change without notice. Prices are U.S.A. only. Calif. residents add 6% sales tax. Prepaid orders shipped postpaid. Bankamericard and Master Charge accepted.

If you haven't seen the POLY 88 yet, why don't you wander down to your local dealer and see what personal computing can be like.

WEST

Bits N Bytes, 679D S. State College Blvd., Fullerton, CA 92631 (714) 879-8386
Byte Shop Arizona, 3237 S. Fairfield Dr., Tempe, AZ 85282 (602) 894-1129
Byte Shop Campbell, 2559 S. Bascom, Campbell, CA 95008 (408) 377-4685
Byte Shop Mountain View, 1063 W. El Camino Real, Mountain View, CA 94040 (415) 969-5464
Byte Shop Palo Alto, 2227 El Camino Real, Palo Alto, CA 94306 (415) 327-8080
Byte Shop Pasadena, 496 S. Lake Ave., Pasadena, CA 91101 (213) 684-3311
Byte Shop Portland, 2033 SW Fourth Ave., Portland, OR 97201 (503) 223-3496
Byte Shop San Fernando Valley, 18424 Ventura Blvd., Tarzana, CA 91356 (213) 343-3919
Byte Shop Walnut Creek, 2989 N. Main St., Walnut Creek, CA 94596 (415) 933-6252
Byte Shop Westminster, 14300 Beach Blvd., Westminster, CA 92683 (714) 894-9131
The Computer Center, 8205 Ronson Rd., San Diego, CA 92111 (714) 292-5302
Computer Components, Inc., 5848 Sepulveda Blvd., Van Nuys, CA 91411 (213) 786-7411
The Computer Mart, 625 W. Katella # 10, Orange, CA 92667 (714) 633-1222
Computer Systems Unlimited, 18886 Hesperian Blvd., Hayward, CA 94541, Hrs. W-F, by appt., S&S 1-7 (415) 278-2667

MIDWEST

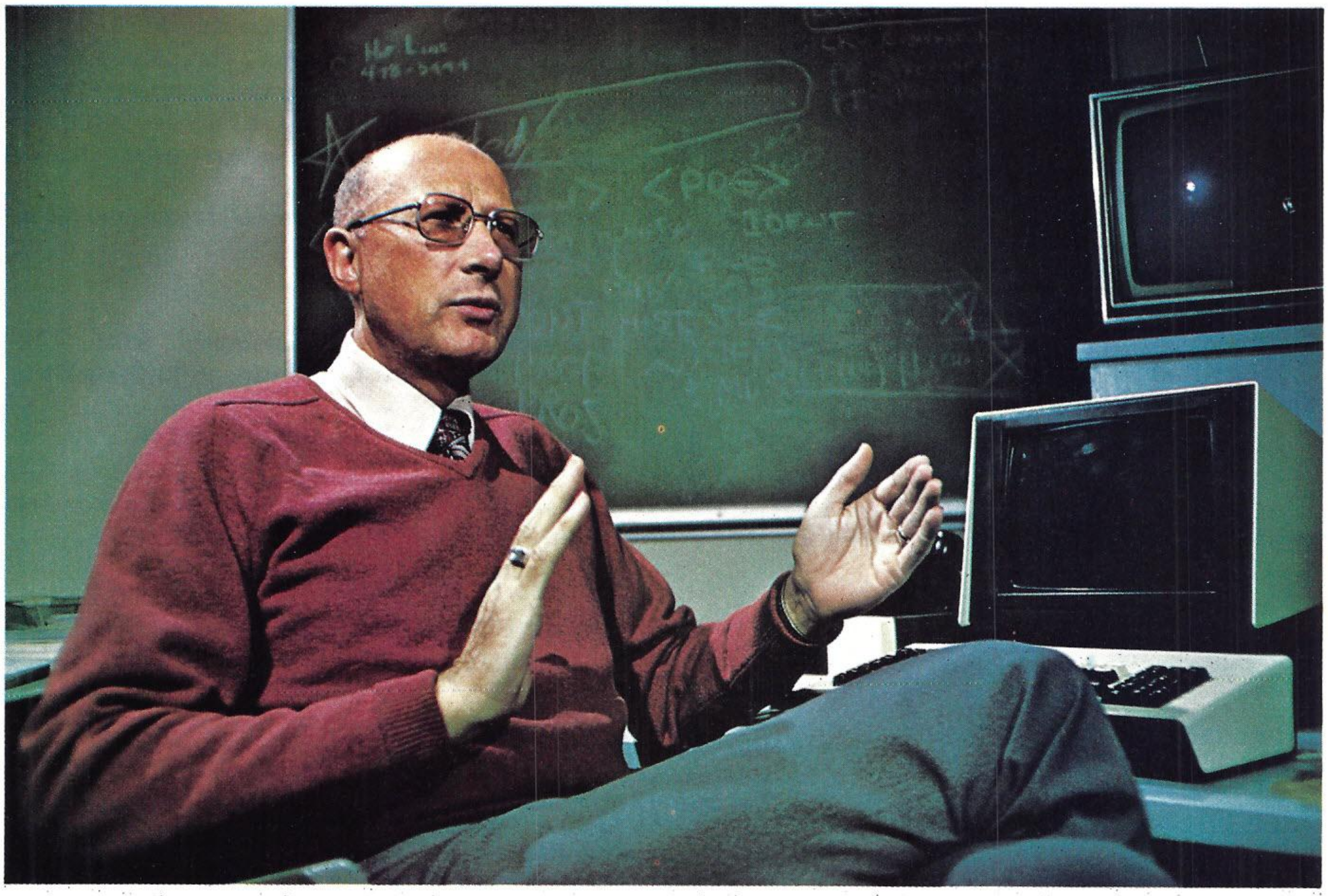
Computer Workshop of Kansas City, 6903 Blair Rd., Kansas City, MO 64152 (816) 741-5050
The Computer Room, 1455 S. 1100 East Salt Lake City, UT 84105 (801) 466-7911
The Data Domain, 111 S. College, Bloomington, IN 47401 (812) 334-3607
The Data Domain, 7027 N. Michigan Rd., Indianapolis, IN 46224 (317) 251-3139
itty bitty machine company, 1316 Chicago Ave., Evanston, IL 60201 (312) 328-6800
The Micro Store, 634 S. Central Expwy., Richardson, TX 75080 (214) 231-1096

EAST

Micro Computer Systems, Inc., 144 S. Dale Mabry Hwy., Tampa, FL 33609 (813) 879-4301/4225
Elecon Corp., "The Computer Store", 4921-72 Ave. No., Pinellas Park, FL. 33565 (813) 541-3021
Computer Shop 11111111, Route 16B, Union, NH 03887 (603) 473-2323
Computer Workshop, 5709 Frederick Ave., Rockville, MD 20852 (301) 468-0455
Hoboken Computer Works, No. 20 Hudson Pl., Hoboken, NJ 07030 (201) 420-1644
The Computer Mart of New Jersey, Inc., 151 Kline Blvd., Colonia, NJ 07067 (201) 574-2173

PolyMorphic Systems
737 S. Kellogg, Goleta, CA 93017
(805) 967-2351

CIRCLE 24



"Generally, you have to assume that passing criminal laws never 'solves' the problem. There's a criminal element in society that's always there, ready to take advantage of everything that comes along, every innovation, every new loophole."

A Personal Computing Interview

Donn Parker on COMPUTER ABUSE

Donn B. Parker is the now-renowned author of *Crime By Computer,* published by Scribner's, a work that has established him as the world's foremost authority in the area of computer crime. This report is excerpted from an interview he granted to PERSONAL COMPUTING in September, 1976 at his office at Stanford Research Institute in Menlo Park, Calif.

PERSONAL COMPUTING: Don't you get a lot of complaints that you are teaching criminal techniques in your book and lectures?
PARKER: I hear some complaint, but I take my cue from the FBI, which publishes a great deal about crime, assuming that potential victims are better off being warned about what the criminals already know.
PC: Are there a lot of variations on computer crime?
Parker: There's something weird, something crazy about every one of these things. We've got over 420 of these cases now, three new ones just last week. In one, a gang stole $14 million worth of gasoline. It was a big thing with trucks and a number of people involved, but one key ingredient to the crime was that they had to change inventory records inside the computer system.

In a second, prisoners at Leavenworth have been learning computer programming and data processing for several years. They have on-line terminals to commercial timesharing services and so on. Apparently prisoners there are now involved in tax fraud. That's somehow being tied in to their computer usage.

And in a third, a Southern California gang was just indicted for charging anywhere from $150 to $1500 to change someone's bad credit rating to a good credit rating. They were doing that through an employee who had access to the computer files.
PC: Does personal computing by individuals open new vistas of crime?
Parker: Yes, of course, but you know, my advice, even to legitimate clients in ordinary business is: Do everything you possibly can to avoid using a computer. As a last resort, if there's nothing else you can do, OK, fine. But computers are a pain in the neck. They're complex and expensive. You really have to have a good application to make them pay.
PC: You think a criminal using a computer has a higher risk of being caught?
Parker: You can dream up all sorts of scenarios in which computers could be useful in crime, but you'd have to be working on some extremely complicated crime before the use of a computer would be worth the extra complication and the chance that it wouldn't perform right.
PC: Can you give an example of a crime complicated enough?
Parker: Yeah. A gang in London decided to carry out a check kite. Check-kiting is where you open several accounts in several banks and you write a check against one account for more money than it really has in it and deposit that in a second account, against which you can write checks that appear to be covered. Sometime, each of us has probably raced to the bank to cover a check that would overdraw our account. Check-kiting is just a formalization of that overdrawing process.
PC: You do it on purpose and run with the money before the banks try to clear the bad checks?
Parker: The problem with check-kiting is its complexity, if you want to do it on any major scale. You've got to have a lot of accounts. You've got to write a lot of checks, keep track of where they are, when they're to come in, and so on. To do it on a grand scale really takes a computer.

This gang leased a minicomputer and got an inventory control software package. All they did was identify the banks as warehouses and the money in them as the content of the warehouses.
PC: Did they have inside information on how long it took checks to clear at which banks?
Parker: Well, they tested all these

things first with legitimate transactions so they had the information. That's what they fed to the computer, telling it the time it took to move inventory around. They had it all figured out.
PC: They simulated the whole thing first?
Parker: Yes. And on the appointed day they sent gang members all over greater London to open accounts and start writing checks, building the kite. They

The embezzler told me he would have been caught soon, anyway, because he just couldn't keep it all going. Embezzlement is very hard work.

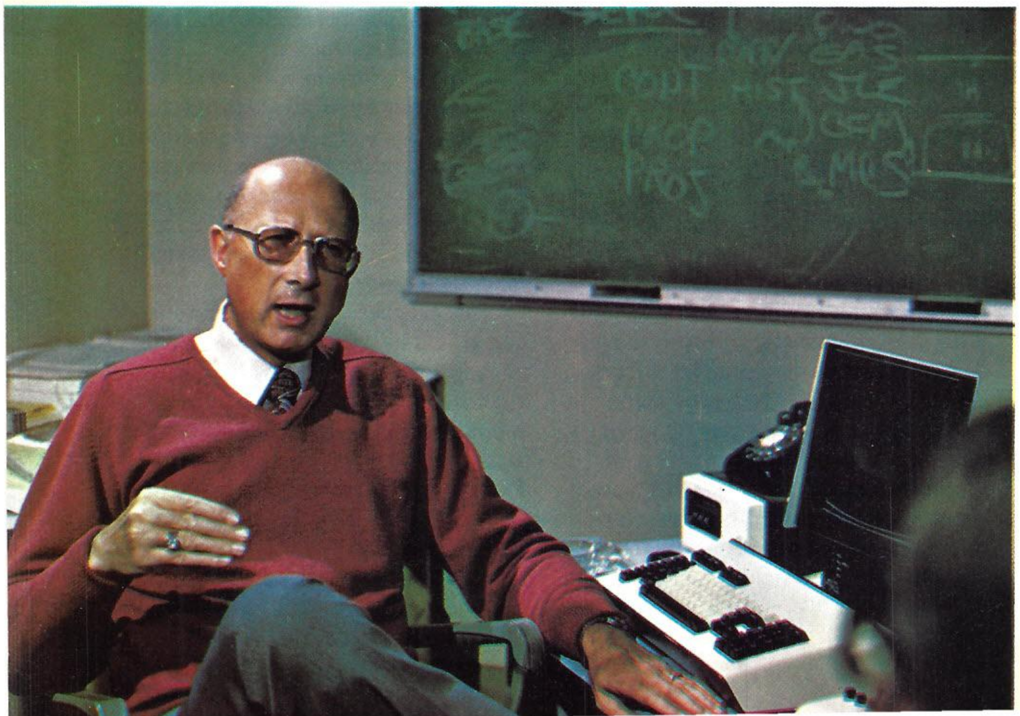
were supposed to get reports out of the computer, showing them where everything was so they could take the next step and the next.
PC: Supposed to?
Parker: Right near the beginning of the thing, but after they'd already gotten into it, their program developed a bug and wouldn't work correctly. Their programmer just went to pieces and couldn't figure out how to get the darned thing to work, so all these people were up in the air with their check kite, waiting for instructions on what to do next. There were no instructions, because the whole system broke down. Next thing they knew, Scotland Yard was running around, grabbing all these people.

They tracked the gang back to their headquarters and found the poor programmer in a state of nervous collapse, panting over the computer.

The computer was returned to the manufacturer – who never did get anything on the lease. There's an example of how a personal computer might be turned to crime. I have lots of cases like that.
PC: Even aside from the moral issues, though, you recommend against such activities on practical grounds?
Parker: Oh, yes. Remember a bank embezzlement case in which the head teller was engaged in a lapping fraud, depositing only part of the customer's money, pocketing the difference, and telling the bank's computer that only the smaller amount had been deposited. The trick is to apply incoming money today to accounts you stole from yesterday and which you expect to be noticed. That's where the "lapping" of accounts comes in. It's a kind of pyramid you're building up.
PC: Isn't it easy with the computer?
Parker: Well, this guy had access to the computer, all right, and was hiding the embezzlement in the computer, because he could move the accounts around fast enough to stay ahead of the auditors. However, he did not have access to the computer to *keep track* of the embezzlement. When he was caught, he had little pieces of paper stuffed in his pockets, his desk, his room, everyplace. He had notes written all over the backs of envelopes, trying to keep track of over 50 accounts that he was juggling. After a couple of years, he was a million dollars in the hole, gambling away the money, hoping to win enough to repay it.
PC: He finally lost control?
Parker: Actually, the police raided a bookie joint and thought it was unusual for this man, an $11,000-a-year teller, to be betting $30,000 a day. With this lead, they investigated. The embezzler told me he would have been caught soon, anyway, because he just couldn't keep it all going. Embezzlement is very hard work.
PC: You talk to a lot of these people?
Parker: Nineteen so far.
PC: Do you find common personaltiy traits among them?
Parker: In fact, part of our research work is to try to identify the profiles of these people, what is common among them. We're learning some things.
PC: Since this field is new, can you clearly define what is criminal and what is not? Could amateurs blunder into crime accidentally?
Parker: I had a little dispute with the FBI recently. They were unhappy with me because I was getting a lot of exposure and claiming all these computer crimes, using some FBI statistics on bank fraud and embezzlement. The FBI pointed out to me that they have a very narrow definition of computer

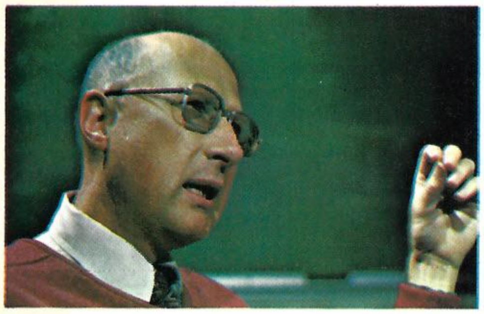
I have found that it is fairly common practice for programmers, especially in timesharing companies, to dip into their competitors' timesharing services.

crime, so narrow that it actually has to be a crime perpetrated inside a computer system for them to count it.
PC: They're talking about things like depositing the odd fraction of a cent in a special account?
Parker: That's right. I would agree that I have really very few of those exotic kinds of things. When I'm speaking loosely to a lay audience. I use the word "crime" as I do in the title of my book, but when I'm trying to be very precise and technical, then I try to avoid the word "crime". Crimes are defined differently in different legal jurisdictions, and everybody has his own definition. In fact, four of our latest reports are on that subject, the adequacy of the law in various areas.
PC: Are new laws needed to solve these problems?
Parker: Generally, you have to assume that passing criminal laws never "solves" the problem. There's a criminal element in society that's always there, ready to take advantage of everything that comes along, every innovation, every new loophole. In a sense, new laws create new opportunities for criminals.



We *have* identified various jurisdictions where the laws are definitely not adequate. I'm thinking now of things like stealing a copy of a computer program from the memory of a computer.

In Texas they put a programmer



away for five years on that, because in Texas they have a very good grand theft law that considers a computer program an asset subject to theft.

In California the law doesn't anywhere describe a computer program as an asset subject to theft, so when a guy here stole a program from memory, they charged him under a new law treating theft of a trade secret, a law that has never been tested. They had to prove that a computer program is a trade secret, which is a very different process than just proving grand theft. The criminal code is not adequate for rapidly advancing computer technology today.

You can easily commit a crime accidentally. Consider the question of smuggling a program that has value past customs by sending it over telephone transmission across an international border. That's a new problem that few people think about.

PC: And computer users don't always realize that what they're doing is wrong?

Parker: Right. For example, throughout the history of computers and computer programs it was just assumed that programs were owned by the people who write them. That's the way the field grew up. In the past few years, top management has begun to understand that programs are extremely valuable assets, the property of their organizations. A lot of programmers still feel that if they write a program, even as an employee, it belongs to them. They feel they own it. There's still a lot of exchange of programs going on that is unauthorized and some that could be defined as crime.

PC: Are computer people out of touch with the ordinary rules of our society?

Parker: There's a tradition, among systems programmers chiefly, of just using computer time for whatever purpose they wish. I have found that it is fairly common practice for programmers, especially in timesharing companies, to dip into their competitors' timesharing services.

Programmers tend to think of computer time as their domain. They have a "right" to computer time because of their technology. It's easy for them to rationalize: "There are the machine cycles not being used. Why in the world don't I use them for some purpose?"

I've run into that as manager of a data center. People want to use the machine at night to run roulette odds, that sort of thing, and if they ask me, I say no. We can't do that. We can't let somebody use at no cost what we charge other people for.

PC: Can personal computers take up some of this slack?

Parker: Some, I suppose. You know, there are people who are completely caught up in computing, whose lives begin and end with the computer. They feel that anything they want to do is acceptable, because they are part of the whole technology. These people *might* be satisfied with personal computers.

PC: Will the diffusion of personal computing throughout society reduce the notion that computer people are a priveleged elite?

Parker: That could work both ways, I think. The people who think of themselves as an elite now may feel a need to prove that they are different from the hordes and seek to achieve greater feats that prove the difference.

I had a meeting with Captain Crunch the other night, John Draper. He's a fascinating guy, the hero of the Blue Box phone freaks. (Do you want the phone number of his PR man who arranges his interviews? You'll have to hurry if you want to catch him.) He's just been convicted of his second offense, and we wanted to get together before he goes off to jail.

In the last few years he has become

It is in the American tradition that a person is free to begin again, to go somewhere else and make a new start if he has begun badly. We're rapidly ruining that.

a computer freak, too. We heard that he was heavily into microcomputing and, given his reputation for knowing all about the telephone system, we felt that here was a potential for a major problem.

He sees himself as scrupulously ethical, serving a higher good by doing illegal things that point out dangerous weaknesses in our communications systems. Extremely immature.

PC: Do you think we're about to see legislation requiring licensing of all computers?

Parker: Well, that would be hard to do. Apart from whether it's right or not, Congress has not been able to pass such legislation with respect to gun control, for example.

One can conceive of the need for a "concealed computer" law. That would make it illegal to engage in any kind of business activity with a concealed computer. I got that from a chapter in my book about the guy with the fastest toes in the west.

PC: Toes?

Parker: He strapped a computer to his stomach and went to Nevada to play games with it. He had a tiny digital

The incidence of computer crime is low and dropping, but the crimes can be big ones, involving big losses.

display wired into his glasses where he could read it, and he controlled the computer with his toes. He said it was really hard to input data fast enough and he practiced a lot, playing license plate blackjack while he was driving, for example.

This is merely a portent of the future, when people can be wired into powerful computers. Imagine business negotiations in a large oil leasing kind of thing. One guy has a hidden microprocessor and the other guy does not. This hidden computer would be a very significant advantage.

PC: Doesn't a smart guy always have the same kind of advantage over a dumb guy?

Parker: Yes, but not to that degree. It's accepted through history that the smart guy has an advantage over the dumb guy, but an artificially produced smartness may be another matter.

PC: There's a lot of worry about "paper people" who assume well-documented false identities to commit fraud. Will personal computing make that easier?

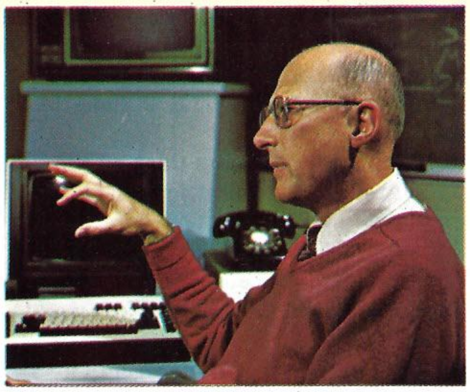
Parker: I don't see how, just offhand. Of course it is in the American tradi-

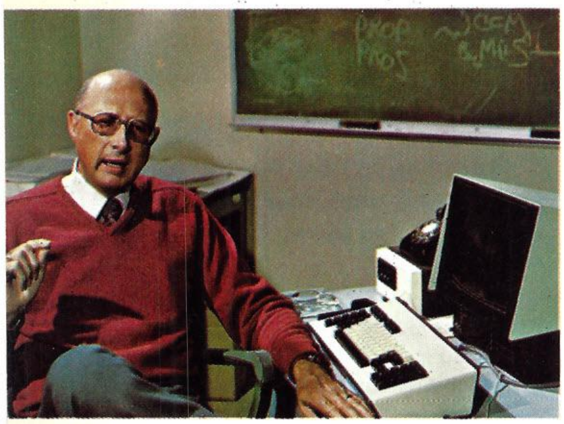
tion that a person is free to begin again, to go somewhere else and make a new start if he has begun badly. We're rapidly ruining that.

Actually, you can hide yourself more effectively today, but it takes more effort to do it. It's like white-collar computer crime, harder to do, but if it is accomplished, the effects are great. The incidence of computer crime is low and dropping, but the crimes can be big ones, involving big losses. I just don't know how personal computers might help paper people.

PC: Is it fair to say that you're making good living on computer crime?

Parker: I think so. Support for the research is not very much, not enough to keep me going exactly full time. Under my National Science Foundation grants I'm supposed to do scholarly papers and publish them. In my SRI work, I'm supposed to provide services to SRI clients. I do both.





Photographs by Paul Honoré

A good thing about an NSF grant is that all the work you do is essentially in the public domain, so my complete file of computer crimes, my notes, and reports are open. Anybody can come in and use my files, and lots of interesting people do.

PC: You've had a series of NSF grants?

Parker: Yes, under the general heading of "Computer Abuse". I just got a grant to hold a national workshop on ethics in the computer field, to study ethical issues that are unique in some way because of compter technology. I'm planning to have 25 or 30 ethical philosophers and people from the computer field.

PC: To belabor the point: you are fairly confident that we're not soon in for a round of laws that will put down personal computing?

Parker: Well, I certainly expect to testify for the Ribicoff committee that it is too early to make effective laws in the area of computer crime. We don't yet know what we want to accomplish.

PC: Your career has been spent working mostly with large computer systems. Are you clearly aware of the cost and capabilities of microcomputers?

Parker: As a matter of fact, my son and I are planning to spend some time in the Byte Shop this Saturday.

PC: You think you'll take a computer home with you?

Parker: Yes, and how it can be used for any illegal or unethical purpose really remains to be seen.

Stanford Research Institute, Menlo Park, Calif. 94025, has published a series of reports prepared for the National Science Foundation under the grants mentioned in this article.

Some of this work has extraordinary significance to those engaged in personal computing, and the reports would be valuable additions to personal or club libraries. There is a charge for some of the reports, so a letter of inquiry is advisable. Recommended:

- Computer Abuse Perpetrators and Vulnerabilities of Computer Systems — December, 1975
- Computer Abuse Assessment — December, 1975
- The Criminal Law Aspects of Computer Abuse: Applicability of the Federal Criminal Code to Computer Abuse — June, 1976
- Criminal Sanctions Under the Privacy Act of 1974 — June, 1976
- The Criminal Law Aspects of Computer Abuse: Applicability of the State Penal Laws to Computer Abuse — June, 1976
- Legal Protection of Proprietary Rights in Software — June, 1976

And, not to leave the matter unclear by default, PERSONAL COMPUTING urgently recommends against formation of any variation of the LEMONADE COMPUTER CRIME SERVICE COMPANY.

FUTURE ISSUES

FUTURE ISSUES
FUTURE ISSUES
FUTURE ISSUES
FUTURE ISSUES
FUTURE ISSUES
FUTURE ISSUES

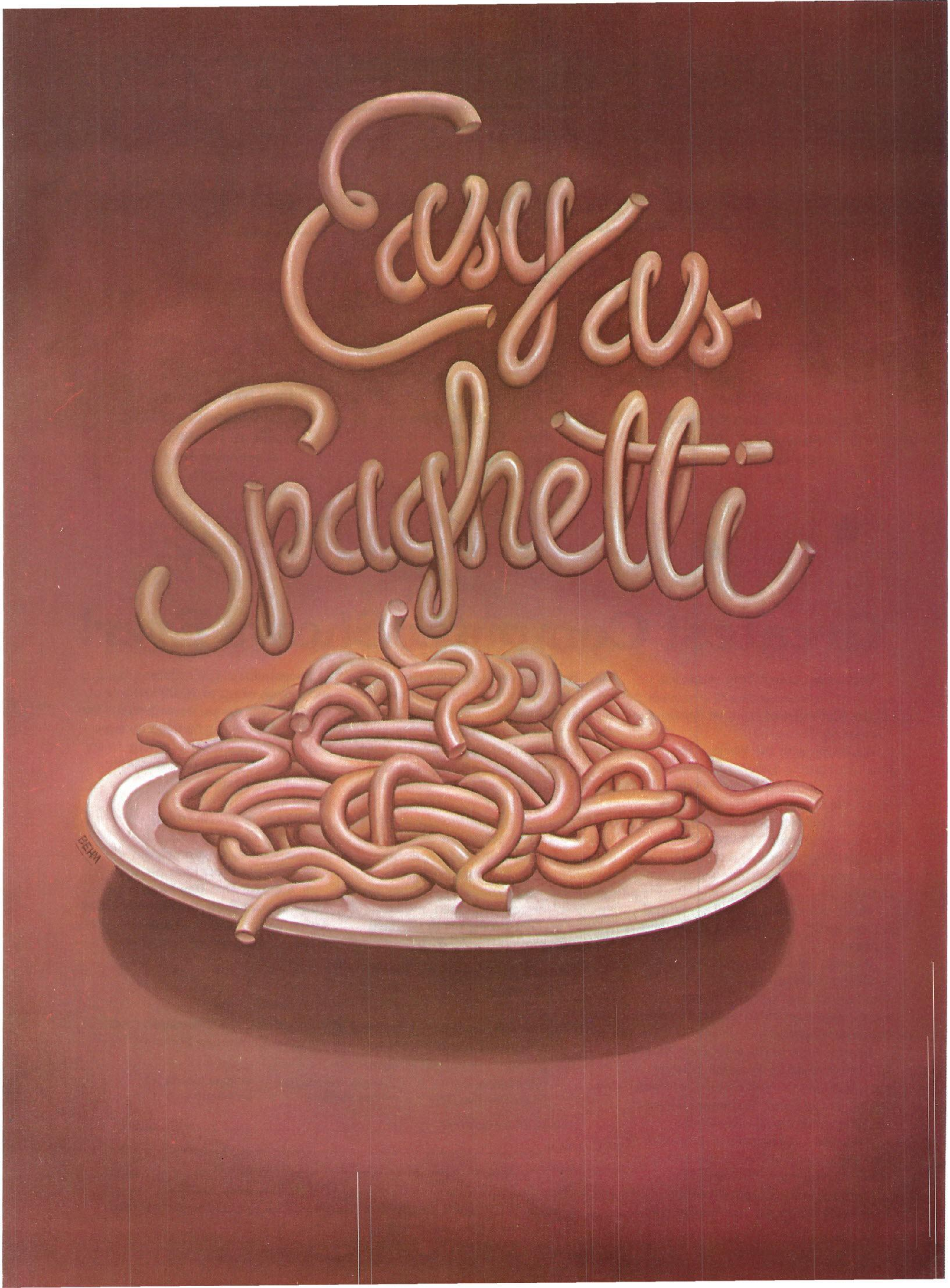
Interviews will be a regular feature of **Personal Computing.** We have contact a number of influential people and hope to have some surprises for you.

Our next issue will have a feature article on computer stores. Nels Winkless will be visiting some of the more interesting stores around the country to report on their progress. In addition, we will be talking to a number of large chain stores to find out it they are interested in selling personal computers. We plan to ask them, "If you had to sell a million computers, how would you do it?"

We have contracted a furniture designer to repackage a computer so that it is more aesthetically pleasing for use in a home. We will provide you with a number of ideas you can use to fit your computer inside your home without making it stick out like a sore thumb.

Our next "Future Computing" article will examine personal computers in schools. Dr. Peter Grimes of the San Jose Unified School District has agreed to provide us a manuscript. His schools have 10 IMSAI systems which they hope will replace their time-sharing system.

Other regular features of **Personal Computing** include "Spaghetti BASIC", "The Software Column", Lemonade Computer Service Company", Random Access" and "Hard Talk about Hardware". Russ Walters has also promised a followup article on "Better Than BASIC".


Easy as Spaghetti
BEHM

SPAGHETTI BASIC

"You gots 70 comin for dindin.
Whaddaya serve em? Spageddy.
Boil way your trubbles.

Ye olde lite blinker gots bout
70 op codes. Whaddaya serve t'it?
BASIC. Tis EASY AS SPAGEDDY.

Put on your specs, Granny.
Here's Papa Dave's cookbook."

Computer Programmer.

Unlike the human brain, a computer is limited to a relatively few simple but powerful processes of problem solving. A computer programmer understands these processes and writes instructions to the computer telling it how to solve a particular problem. These instructions are called a program.

From my limited observation of computer programmers, I must add that they're typically night owls. It is not at all unusual to walk into a computer company, say at 9 a.m., and find a computer programmer fast asleep, draped over a CRT terminal or a Teletype. He's been there since 2 p.m. the previous day and if you don't disturb him chances are good he'll wake up in a few hours and immediately continue his work as if nothing happened.

Definition: A program is a set of instructions to a computer telling it how to solve a particular problem.

Four Parts to Writing a Program

There are only four parts to writing a program. Once you have learned some of the basic programming skills you will find that the hardest part is the first part, namely: DEFINING THE PROBLEM. The other three parts, INPUTTING THE DATA, COMPUTING THE DATA and OUTPUTTING THE ANSWERS, depend on knowledge of programming techniques. The more techniques you are familiar with, the easier it gets.

Just How Easy Is It?

Consider the problem of figuring *add-on,* or as it is better known, simple interest rates. Let's say you recently purchased a motorcycle for $750. You made a down payment of $100 and financed the rest of 18 months at an add-on rate of 6½%. You probably want to know the answers to three questions:

1. What is the total amount of interest you are going to pay?
2. What is the total amount of money owed?
3. What are your monthly payments going to be?

There, we just did the hardest part. We DEFINED THE PROBLEM.We determined exactly what we want to know about this loan.

Inputting the Data

As previously stated, a computer doesn't think – it merely manipulates data. Therefore, a program must supply the computer with the data to manipulate:

1. Principal of the loan – $650.
2. Rate of interest – 6½%.
3. Time – 18 months.

Computing the Data

Once we have entered data into the computer, we have to tell it what to do with the data.

From elementary arithmetic we know that the formula for figuring add-on interest is: *Interest* equals *Principal* times *Rate* times *Time* or I=PxRxT. Since time in this formula refers to the *number of years* and we have entered 18 months, we have to divide our answer by 12. The formula then reads: I=PxRxT/12. To find the interest on our loan we *instruct the computer* to multiply *Principal* times *Rate* times *Time* divided by 12.

In addition to the amount of interest, we want to know what our *total payment* will be. The formula is *Total Payment* equals *Principal + Interest* or (letting Pl stand for *Total Payment)* Pl=P+I. To find the *Total Payment* we instruct the computer to add the *Principal* to the *Interest.*

The third and final question we want answered is: what is the amount of our *monthly payment?* The formula is *Monthly Payment* equals *Total Payment* divided by *Time* (where *Time* is in months) or M=Pl/T. To find the *Monthly Payment* we instruct the computer to divide the *Total Payment* by the *Time.*

Outputting the Answers

Computers are simple minded. They do exactly what we tell them to do, no more and no less. So, unless we tell

the computer what answers we want it to output (display on our CRT or print out on our Teletype) the computer will keep the answers to itself.

In the fourth part to writing a program, OUTPUTTING THE ANSWERS, we have to tell the computer to output the three answers we want:

1. Total interest
2. Total amount owed
3. Monthly payment

The Program

The following program answers our three questions. While at this point it may be obscure to you, the program will become clear once you are familiar with its various components.

```
NEW
10 LET P=650
20 LET T=18
30 LET R=.065
40 LET I=P*T*R/12
50 LET PI=P+I
60 LET M=PI/T
70 PRINT "TOTAL INTEREST IS"; I
80 PRINT "TOTAL MONEY OWED IS"; PI
90 PRINT "MONTHLY PAYMENTS ARE"; M
RUN
```

BASIC Elements

BASIC language offers only two formats for inputting data into the computer. These formats are *statements* and *commands.*

The line number preceding a statement tells the computer not to execute. Execution begins only when a command is given.

A *statement* is an instruction to the computer. It is executed only after a command has been given. The first statement in our program, 10 LET P=650, instructs the computer to assign a value of 650 to P. As you may recall, 650 is the principal on the loan.

BASIC language contains a number of different kinds of statements. Our program contains two kinds of statements: LET statements and PRINT statements. We'll come back to this.

Commands

A command is a direct order to a computer that is executed immediately upon entry. The two commands in our program are NEW and RUN.

Line Numbers

A line number, placed at the beginning of each statement, does two things. First, it tells the computer not to execute the instruction that follows until a command has been entered. Second, it gives order to the execution of the statements in a program. Following the entry of a command, the first statement to be executed is the one with the smallest line number, then the next smallest and so on.

The order in which statements are entered into the terminal has no meaning. Had the first three lines of our program been entered into the terminal:

```
20 LET T=18/12
10 LET P=650
30 LET R=.065
```

they would be executed in exactly the same order as the first example, beginning execution at line 10.

Line numbers can typically be any integer (no decimal) between 1 and 9999. You should leave unused numbers between statements for inserting new statements into a program once it has been written.

Return

Each statement and each command in a program is ended by depressing the RETURN key. When the computer receives the RETURN entry, it does two things. First, it stores the statement in memory. Second, it searches the statement for errors. If an error is found, the word ERROR is displayed, followed by the number code of the error.*

Errors and Spaces

Most errors during statement entry arise from spelling or typing errors. For example, if you entered the first statement of our program as:

```
10 LEET P=650
```

the computer would respond with

```
ERROR 02 – SYNTAX
```

To correct the syntax error, re-enter the line number and retype the statement in its corrected form. The corrected statement replaces the incorrect one because it has the same line number and was entered more recently.

Error numbers and codes are generally explained in the back of the manual that comes with most BASIC languages. We will examine them in more detail in a future installment of Spaghetti BASIC.

Spaces are not required to separate any of the characters in a statement. However, spaces make the statements much more readable and easier for you to find errors.

LET Statements

The first six statements in our example program to solve an add-on interest problem are LET statements:

```
10 LET P=650
20 LET T=18
30 LET R=.065
40 LET I=P*T*R
50 LET PI=P+I
60 LET M=PI/T
```

The LET statement is used to assign a number called a value to an abbreviated code called a variable. As you see, a LET statement can contain a single number of the result of a computation. Note in Statement 40 that the symbol for multiplication is an *. We use an asterisk because a computer is not bright enough to figure out the difference between the letter x and the multiplication symbol ×.

Once a value has been assigned to a variable, the variable can be used to represent the value throughout the remaining program statements. Thus in statement 40, the variable P still represents the value 650.

The word LET in a LET statement is optional:

```
10 LET P=650
```

is exactly the same as:

```
10 P=650
```

Notice that the first three LET statements in our program represent the second part of writing a program, which is INPUTTING THE DATA:

```
10 LET P=650
```

As previously mentioned, this LET statement assigns the value 650 to the variable P. $650 is the amount of principal of the loan.

```
20 LET T=18
```

This LET statement assigns the value 18 to the variable T. 18 months is how much time you have to pay off the loan.

```
30 LET R=.065
```

This LET statement assigns the value .065 to the variable

R. Your rate of interest, 6½%, converts to the decimal .065 (computers like decimals).

Notice also that the next three LET statements represent the third part of writing a program, which is COMPUTING THE DATA:

40 LET I=P*T*R/12

This LET statement assigns the value P*T*R/12 to the variable I. Interest on your loan is equal to principal X time X rate. We divide this answer by 12 because the value we have entered for time is 18 months and the interest formula calls for time to be in years.

50 LET PI=P+I

This LET statement assigns the value P+I to the variable PI. The TOTAL PAYMENT is equal to the principal plus the interest.

60 LET M=PI/T

This LET statement assigns the value PI/T to the variable M. The monthly payment is equal to the total payment divided by time (the number of months).

String Literal

A string literal is a "string" of characters enclosed within quotation marks in a PRINT statement. It instructs the computer to print out the "string" of characters just as they are in the PRINT statement. In other words, to print them literally.

A string literal is generally limited to 72 or so characters depending upon which BASIC you are using.

Print Statements

The last three statements in our example program are PRINT statements.

70 PRINT "TOTAL INTEREST IS"; I
80 PRINT "TOTAL MONEY OWED IS"; PI
90 PRINT "MONTHLY PAYMENTS ARE"; M

A PRINT statement can instruct the computer to display a value or a "string literal". The PRINT statements in our program tell the computer to do both.

The computer prints a space after each number and a space before each number that's positive. In Altair BASIC the word PRINT can be replaced by a question mark.

Notice that these PRINT statements represent the last part to writing a program, which is OUTPUTTING THE DATA.

New and Run Commands

The word NEW, followed by a RETURN, commands the computer to clear its memory of any previously entered data. The purpose of this command, which is generally given at the beginning of each program, is to prevent the computer from confusing a previous program with the one to be entered.

The word RUN, followed by a RETURN, commands the system to execute the program.

Try Some Changes

If you have access to a computer with BASIC language, enter and run our example program. Change the figures and run the program again (you may want to be more realistic, I mean, 6½% interest, really!) Leave out the word LET in the LET statements and substitute a semicolon or question mark for the word PRINT in the PRINT statements. In the next issue of PERSONAL COMPUTING, we'll examine a handy tool for writing programs called "flow diagrams" and we'll examine ways of making our example program more useful with the use of READ and DATA with GOTO.

* In some BASICs, including Altair BASIC, error messages are displayed only after the entire program has been entered and the computer has received the RUN command.

Spaghetti BASIC summarized

In Lesson 1 of Spaghetti BASIC, you have learned the following:

- A program is a set of instructions to a computer telling it how to solve a particular problem.
- A statement is an instruction to a computer.
- A line number is an arbitrary number at the beginning of each statement in a program. It has no significance except that it gives order to the execution of the statements in a program.
- A line number can be any integer between 1 and 9999.
- Each statement is ended with a RETURN.
- Most errors that occur during statement entry are due to spelling or typing errors. Error numbers and codes are generally explained in the back of the manual that goes with BASIC languages.
- Spaces are not required but they are useful.
- The LET statement is used to assign a value to a variable.
- A value can be a single number or the result of a computation.
- A variable can be used to represent any value.
- A variable in a program will retain the value assigned to it unless the value is changed.
- The general form of a LET statement is:

  line-number LET variable = value
- The word PRINT in any PRINT statement can be replaced by a semicolon (;) or a question mark (?) depending upon which BASIC you are using.
- A PRINT statement can tell the computer to print a value or a "string literal".
- A "string literal" consists of a "string" of characters enclosed within quotation marks. It tells the computer to print out the "string" of characters just as they are in the PRINT statement.
- The general form of a PRINT statement is

  line-number PRINT "string literal" and/or value
- The word RUN, followed by a RETURN, commands the computer to begin execution.
- The word NEW, followed by a RETURN, commands the computer to clear its memory of any previously stored data.

(continued)

Some help for the innocent bystander

An elegant discussion of computer language, like this series on Spaghetti BASIC, is no help at all if you don't already know what computer languages *are* . . . or for that matter, what computers are.

A few hints may be useful to beginners here.

Start with the computer.

Indeed, start inside the computer with the Central Processing Unit (CPU).

The CPU in any computer can do only a surprisingly small number of jobs. It can transfer a code made up of ones and zeroes from one place in the computer to some other place. It can combine one code with another, effectively adding or subtracting the numbers the codes represent. It can do a handful of such little chores, logical operations upon which the whole, mysterious structure of computer science is based.

You have to tell the CPU what you want it to do.

Literally, you must set switches in coded patterns that turn on the CPU to do particular tasks. The CPU in a typical microcomputer these days can do a specified task very rapidly, in about 2½ millionths of a second. Thus, when you have set the switches for the first task, the CPU does what you have instructed it to do and is ready for another instruction 2½ microseconds later. You won't be ready to give it another instruction, you slow-poke. The CPU will wait a long time, by its standards, before you give it another command.

The great magic of computers lies in the ability of the systems to re-set themselves automatically, accepting commands that are generated internally by the computers themselves.

Suppose that you have instructed the CPU to transfer a code from one location to another. It does it.

When the code arrives in the new location, that arrival may in itself trigger a command to the CPU to do something else. Thus you can work not only with the information in that code you have moved but with *the information that the code has been moved.* The action continues automatically this way until all of the logical reactions have occurred. It's like knocking down the first domino in a line. The action continues until nothing else *can* happen.

The computer, unlike the line of dominoes, can sustain the action for a long, long time, by setting 'em up again automatically. You don't have to throw the switches yourself 400,000 times a second.

You *start* by setting the command switches in a pattern that will put the computer through a series of automatic steps in processing some information.

You *finish* with a pattern of switch settings that are the product of the computer's activity after it has done all it can do and nothing more *can* happen. (There's no way to knock down any more dominoes.)

The final pattern is a code that may be translated automatically into a word message printed out by an automatic typewriter or drawn out as a picture on a plotter.

To those of us who can barely write our own names and addresses with a typewriter without making errors the very idea of flipping those switches in accurate patterns hundreds or thousands of times is horrifying. Addressing the CPU this way is called "using machine language." It's efficient, from the computer's point of view, but exhausting and infuriating to the casual computer user. (Numbers of fanatical hobbyists enjoy this direct, efficient control of their systems. To each his own.)

Early in the game, however, the experts took pity on casual computer users and developed easier ways to address the CPU with coded instructions.

First, a highly trained computer programmer is persuaded to speak to the computer in machine language. He teaches the computer to recognize simpler, more general statements that will be made later by the casual user.

For example, the computer may require a dozen machine-language codes in a particular sequence to carry out the multiplication of one number by another. The skilled programmer throws all the right switches for multiplication, but then also sets up the computer so that it will remember all those commands in proper sequence when the casual user switches in just one special code number.

That lifts some of the curse for the casual user, but some of us can't comfortably use even a small batch of numerical codes that trigger sequences of action in the CPU. We need more help.

So, the skilled programmer goes another step. He sets the switches on the computer so that when the casual user types the word "multiply" on an electric typewriter connected to the computer, the computer recognizes the word as the equivalent of the numerical code that reminds it to do the dozen operations in multiplication.

This approach makes the computer operate less efficiently than it can with straight machine-language instructions, since it must translate "multiply" to a code that is in turn converted to a string of commands. Further, the stored information for all this remembering and translating uses up an appreciable part of the computer's overall storage capacity. However, the casual user is delighted to use a language that means something straightforward to *him.*

Diligent, skilled programmers in the past 25 years have thoughtfully developed these special codes into powerful languages that allow us to put computers through immensely complex sequences of logical operations with a few simple commands. These codes are called "high-level" languages.

BASIC is such a high-level language, employing what seems like plain English laced with a few extra symbols in a rigid grammatical structure. FORTRAN and ALGOL (remember the opera they used to run for computer people on television every Christmas — *Algol and the Night Visitors?)* are high-level languages, as are SNOBOL, COBOL and LISP among hundreds of others.

Each type of computer differs slightly from all others, and a special dialect of a high-level language is developed for each case. All dialects of BASIC are much alike, so that the ordinary, not-too-skilled speaker of basic BASIC can use any of them easily.

By the way, a skilled programmer doesn't come with your computer in a box from the factory. He doesn't have to be present in the flesh to punch in the basic language codes. You can do that yourself, following instructions, or you can run punched paper tape or magnetic tape to load the computer with the high-level language you want. This is faster and easier to repeat. And the tape reader doesn't demand sandwiches and coffee all the time the way a programmer does.

These Spaghetti BASIC articles suggest attractive ways to use BASIC with a minimum of fuss, once it's in the computer.

ROBOTS

AN ESTIMATE OF THE STATE OF THE ART, AND AN INVITATION TO PERSONS OF ADVENTUROUS SPIRIT AND INQUIRING MIND

We believe that the key discoveries necessary to the art of robotics have already been made. **We believe** that behind various national borders, behind the doors of various scientific disciplines from biochemistry to microelectronics, all of the primary technical obstacles have been overcome, all feasibilities been proven, all methods become known.

We believe that what remains to be achieved is principally the refinement of systems applying existing technologies — **and that this work proceeds apace.** We believe the world is about to encounter (where? when?) machines that truly simulate the intellectual and physical behavior of human beings: robots.

Robots are on our doorstep. Robots are almost within our reach. And we within theirs.

Robots are as frightening as they are alluring, as threatening as they are promising. Yet whatever reservation anyone may feel, there is now no turning back, no possibility of their denial or prohibition. The development of artificial intelligence proceeds not only in the laboratories of governments and industries, but also among the thousands of individual amateurs and hobbyists, free citizens exercising their freedom with experiments in the fascinating field of personal computing. **We believe** that since they are possible, robots are inevitable — "for good or ill."

The United States Robotics Society is established "for good" — for the good of mankind — not in opposition, for opposition is idle, and not in advocacy, for advocacy is unnecessary. **We invite** the support and active participation of all persons who can face the Age of the Robot with the appropriate curiosity and spirit of adventure.

Intelligent machines for production and service — tireless, able to understand commands and carry them out sensibly without feeling a need to make policy for themselves — may become the long-heralded boon to humanity, lifting ancient burdens of toil and suffering. But if they were to be developed "in the dark" — if they were to be sprung upon us full-blown, without our preparation — the reaction might be disastrous. The survival of our own society may depend quite soon (how soon?) on our ability to deal even with "friendly" robots. If we ignore them, if we are incompetent in their fields, we are surely not serving our own interests.

Intelligent weapons now appear practicable within the next decade or two — systems, for example, that can differentiate between friend and foe automatically, through their own sensors and judgement. If such weapons are developed anywhere in the world, they will be extraordinarily dangerous to any society which has not learned how to deal with them.

Robotics has charm not only for trained technicians and professionals but also for millions of persons without the skills and resources to participate directly in the work. Communication about robotics, like robots themselves, is inevitable, — through publicity, rumor, espionage, and now through The United States Robotics Society. This organization will assume the important task of identifying discoveries, gathering supporting data from the hidden recesses where they rest, collating, publishing, becoming a center of information for all parties seeking knowledge of current and historical activity in robotics. We urge you to be one of us — for just $12/year.

Benefits to USRS Members Growing Year by Year

- Certificate of Registration as USRS Member.
- USRS Newsletter, USRS bulletins, other correspondence from the Society as occasion demands.
- Aid in contacting other USRS Members in home regions, toward establishing USRS events.
- Opportunity (Qualified) to officiate as USRS Representative at Regional and National robotics shows and exercises.
- Service (Optional) as USRS Contributing Correspondent.
- Participation in the determination of procedures for investigating, reporting, archiving, and disseminating information relevant to robotics . . . and
- Privileged access to the Library of Robotics to be established by USRS.
- Discounts as may from time to time be arranged by USRS on behalf of members — with publishers, manufacturers/distributors of robotics related materials. (Note: this benefit alone can be expected to repay the moderate USRS Membership costs many times over.)

Coming: Waldo — The Game of Tomorrow's Superbowl.

United States Robotics Society

A Non-Profit Organization
Glenn R. Norris, President
Box 26484 Albuquerque, New Mexico 87102

Application for Charter Membership

USRS

United States Robotics Society
Box 26484 Albuquerque, NM 87102

Enclosed is my check for $12 for enrollment and first-year dues.

The following information is requested (OPTIONAL) to help ensure your full participation in the benefits of the Society.

My interest in robotics derives from ( )intellectual curiosity ( )academic training ( ) professional/business

Please tell us more: ____________________

( ) I am interested in joining with others in local USRS activity.
I might serve as ( ) Correspondent ( ) Official at USRS functions.

NAME ____________________
ADDRESS FOR USRS COMMUNICATIONS ____________________ CITY __________ STATE __________ ZIP ________ PHONE ________


"TICKLE", "ME"
?
"TICKLE" : "ME"
HIGGINS

LET'S IMPROVE BASIC

by Russ Walter

The computer language used most is COBOL. Second is FORTRAN. Tied for third place are PL/1 and BASIC.

Although ninety percent of all programming is in COBOL, sixty percent of the small amount done by high school students is in BASIC, according to casual surveys. University departments of computer science usually start their majors on FORTRAN or PL/1.

Though the easiest to learn, BASIC has glaring deficiencies that make computer professionals shun it. BASIC should be fixed, but before we flourish our scalpels let's see why BASIC became the way it is.

How BASIC arose. John Kemeny and Thomas Kurtz invented BASIC at Dartmouth College, in 1963, by simplifying the computer language ALGOL. They retained the ALGOL symbol ↑ for exponents and the ALGOL words FOR, TO, STEP and THEN but eliminated ALGOL's complex options such as integers versus reals (BASIC automatically gives you reals), local versus global variables (BASIC automatically gives you globals), and simple versus compound statements (BASIC forces you to use simple statements).

Then Kemeny and Kurtz added features ALGOL lacked such as string variables and the MAT package. Though string variables are essential to BASIC's success, the average programmer rarely uses the MAT package's IDN, INV, DET, CON and TRN. Kemeny included them because he needed them for his specialized research on Markov chains.

BASIC became popular for several reasons. It was easy to learn. It used an exciting interactive timesharing system. It compiled quickly. It included four simple pairs of job-control commands: HELLO & BYE, NEW & OLD, SAVE & UNSAVE and LIST & RUN. For editing, it used the statement numbers: "to delete a line, just type the line's number; to correct a line, just retype it." Kemeny and Kurtz mastered politics: they procured funds from General Electric Co. and the National Science Foundation, motivated their students to write the compiler and ran an in-service program for high school teachers, who spread the good news. But the big clincher came when Digital Equipment Corp. decided to put BASIC on its PDP-8 computers as an alternative to FOCAL. PDP-8 users preferred BASIC to FOCAL so much that Digital chose BASIC to be the main language for its PDP-11.

As BASIC's popularity spread, people wanted to apply the language to more complicated problems, so Kemeny and Kurtz had to add more features. What we have now is an ideal tiny language with a lot of fancy features stacked on top.

Why BASIC isn't tops. Suppose you try to reach the highest point in the world by always walking uphill. Though you'll wind up higher than you started, you won't reach the top of Mt. Everest: to reach that pinnacle, you must occasionally walk downhill to get to a higher mountain.

Unfortunately, Kemeny and Kurtz always walk uphill: they adopt a new feature only if it agrees with programs written previously. Following their plan, all the wrong steps we've taken during the past dozen years will never be reversed; all future generations will suffer.

It's time to get off our hill and go climb a mountain. Here's how.

BASIC handles the word "to" inconsistently. In a LIST command, most versions of BASIC indicate "to" by a dash (LIST 100-500), but in a FOR command it is spelled out (FOR I = 1 TO 10). Let's be consistent! Let's follow the habit of most other computer languages and use a colon, saying LIST 100:500 and FOR I=1:10. Let's use the same notation for substrings: A$[3] is the 3rd character of A$, but A$[3:7] consists of characters 3 to 7. Omitting the number after the colon should indicate infinity.

STATEMENT	MEANING
`LIST 100:`	list line 100, and every line thereafter
`FOR I=1:`	for I = 1, 2, 3, etc., forever
`A$[3:]`	A$, but begin at the 3rd character

BASIC has a confusing display of "end" words: a FOR loop must end with NEXT; a subroutine must end with RETURN; a multi-line function definition must end with FNEND. Let's use a single method: indent. Instead of our writing –

```
FOR I = 1 TO 3
PRINT "CAT"
PRINT "DOG"
NEXT I
PRINT "GROWL"
```

– let's write:

```
FOR I=1:3
    PRINT "CAT"
    PRINT "DOG"
PRINT "GROWL"
```

Indenting each FOR loop, subroutine and multi-line function shows more clearly where each structure begins and ends (especially if there are loops inside loops inside loops) and eliminates the novice's error of writing "overlapping loops", "FOR without NEXT" or "NEXT without FOR". Kemeny and Kurtz themselves recommend indenting. By abolishing the words NEXT, RETURN and FNEND, we can force the programmer to follow Kemeny and Kurtz's recommendation.

BASIC forces you to think negatively. Instead of thinking, "if N is less than 3, do the following 10 statements", you must think "if N is greater than or equal to 3, skip the following ten statements", and write it like this:

```
IF N>=3 THEN s
```

– where s is the number of statement following the 10 statements you want to skip.

That negative thinking confuses the beginner. To say "if

HE'S GOT LOGGYRITTEM! RANDOM RAN RA!

N<3, increase both J and K by 1", he is inclined to write –

```
100 IF N<3 THEN 110
110 J=J+1
120 K=K+1
```

– but the correct negative solution is:

```
100 IF N>=3 THEN 130
110 J=J+1
120 K=K+1
```

Once again, indentation comes to the rescue. Let him write what he thinks:

```
100 IF N<3
110     J=J+1
120     K=K+1
```

The computer performs the indented lines if N<3. Using indentation, we can eliminate the word THEN from the language. It was an awful word anyway; IF N>=3 THEN 130 is less grammatical than IF N>=3 GO TO 130.

Let him write "if N<3, increase both J and K by 1; otherwise, multiply both J and K by 2" like this:

```
100 IF N<3
110     J=J+1
120     K=K+1
130 IF NOT
140     J=J*2
150     K=K*2
```

BASIC's insistence on short variable names makes programs hard to read. COBOL and PL/1 let you say CIRCUMFERENCE = 7.5, and FORTRAN lets you say at least CIRCUM = 7.5, but in BASIC you must say C = 7.5; it's hard for the reader to tell whether the C stands for circumference or centigrade or count or calories. Let's allow long variable names; but to quicken the compiler, let's make it look at only the first two characters, so it reads CIRCUMFERENCE as CI. To be clear, the programmer can say CIRCUMFERENCE at the beginning of his program; to be brief, he can say CI later; the computer will understand they are the same.

Teaching the computer to examine only the first two characters lets the programmer say PR instead of PRINT, SQ or SQRT or SQUAREROOT instead of SQR, LOGARITHM or LOGGYRITTEM or LO instead of LOG, and SINE or SI instead of SIN. BASIC's unpronouncable RND should be renamed RANDOM, which can be abbreviated RA or RAN.

Since PR IN is supposed to print IN and PRI N is supposed to print N, we want the computer to distinguish PR IN from PRI N, so we must tell the computer to notice spacing.

BASIC's PRINT statement contradicts English. In normal English, a semicolon indicates a longer pause than a comma; but in BASIC a semicolon indicates *less* spacing. Let's follow English: a comma should indicate little spacing, a semicolon should indicate more and a colon should indicate the most. Specifically, *a comma should indicate no spacing at all, a semicolon should indicate a single space and a colon should indicate tabbing to the next 14-character zone.* Let's eliminate BASIC's hard-to-remember rule that a space follows each number but not each string and that a space precedes each positive number but not each negative.

STATEMENT | **WHAT IT PRINTS**

```
PRINT "HOT","DOG";"MAN"  HOTDOG MAN
PRINT "TICKLE":"ME"      TICKLE        ME
PRINT 2+2,6+3;8;-5;;;-7  49 8 -5    -7
```

If X$ is a person's name (such as "JOE"), and N is his score on a test (such as 98 or -5), and you want to print a sentence such as TODAY JOE SCORED 98 or TODAY JOE SCORED -5, this statement should do it:

```
PRINT "TODAY";X$;"SCORED";N
```

Simple, isn't it? By contrast, to get such good spacing in traditional BASIC, you'd have to put spaces in quotation marks and test N's sign:

```
100 PRINT "TODAY ";X$;" SCORED";
110 IF N>=0 THEN 130
120 PRINT " ";
130 PRINT N
```

BASIC contains useless junk. Let's sweep out the cobwebs by eliminating the words END, LET, REM, ON...GO TO, SGN, IDN, CON and TRN. Eliminating END and LET is hardly a new idea: PDP-11 computers have been ignoring these words for years. REM was a great idea when it was invented, but Dartmouth has subsequently discovered a better way to insert a remark: put it on the tail end of a normal line after an apostrophe. This so-called "apostrophe convention" eliminates the need for REM. The ON...GO TO statement was an essential tool (especially for printing strings) when BASIC was weak, but the later introduction of string arrays, the new concern for structured programming and our powerful indented IFs make it obsolete. The SGN function is nowadays used only in the ON...GO TO statement, so we can throw it out also. IDN, CON and TRN are used only by specialists analyzing Markov chains and can be easily programmed by hand.

Many other features of BASIC need improvement (especially subroutines), but I'll shut up until the next issue. I don't want to poison you with my ideas. How would *you* improve BASIC? Please tell us. If your comments are interesting or useful, we'll publish them.

Russ Walter *is Assistant Editor of* Personal Computing *and* Minicomputer News *and author of* The Secret Guide to Computers, *a five-volume set introducing the novice to computers; the first volume is on BASIC. He heads the computer program at Wesleyan University's Graduate Summer School for Teachers, where he trains high school and college teachers to give lessons on BASIC and other languages. He has taught BASIC to 2nd grade children, retired persons and every age in-between.*

He and BASIC grew up at Dartmouth College together, where he studied under BASIC's inventor, John Kemeny. From watching many people try to learn BASIC, he has concluded that BASIC would be easier to learn if it were slightly revised. This article expresses his opinions. What are yours?

By Paul Allen
Vice President, Microsoft

Since my associates and I in Microsoft wrote Altair BASIC, we find that we are almost "old-timers," strange historical figures behind one of the first major personal computing software systems. At this point, I suppose we have as much experience in dealing with the special interests and concerns of non-professional computer-users as almost anybody in the world. It is surprising to realize how new the field is. I was in high school in the late Sixties, and I don't feel like an historical figure. But things do move along pretty fast.

As Bill Gates and I share writing this column over the next months or years, we'll try to give some insights into software that are especially useful.

In fact, we can already see some interesting changes developing that alter our original view of personal computing software. Dramatic improvements in software in the foreseeable future depend on changes in hardware. The hardware inevitably leads. An example:

When we wrote Altair BASIC, our goal was to make it efficient, to pack as much as we could into as little memory as possible, because personal computing systems were characteristically short of memory. Every byte we could save was significant.

In fact, I think we did a fair job of packing BASIC efficiently. As we study our original work now with a view to adapting it to microprocessors, for example, we find that even with the advantage of hindsight and experience we can't make a great deal of improvement.

In the meantime, and this is the major point, the cost of memory has dropped dramatically. The cost of 16K of memory a couple of years from now will probably be down to what 4K costs today. The priority has shifted from efficiency to high performance. It's hard to say exactly how hardware changes of this sort will affect software, though we have a general idea of progress that is planned and hoped-for.

We find that people don't usually realize how powerful their home computers are. Often, for example, students or businessmen have access to computer systems at school or work and play with them or use them for minor tasks.

They don't realize, often, that these are timesharing systems or setups that are otherwise limited. They get only a small part of the computer's capabilities. They are surprised to discover how powerful their home computers can be.

Much of what we do is directed to making these capabilities available to the surprised user. We chose BASIC originally, not because it seemed to be the best possible language that could be developed but because it was already in common use. Much literature was already available, and almost anyone could find some way to educate himself to the use of BASIC. We want to make it easy for users to acquire the skills they need to put their computers to work.

Initially, the proud owner of a new hobby computer may stare at the computer in consternation, expecting it to *do something,* while the computer merely stares back with an ironic glint in its front panel lights. First programs, such as tests, usually only verify that the hardware is performing correctly.

Soon the user is playing games, often of a gambling nature (blackjack, roulette, poker, etc.), or games usually played between two people, with the computer substituting for one of the players (sticks, checkers, chess . . .). Sometimes the computer is used to generate a new backdrop for a combat situation between two people (Spacewar) or between a player and the computer which is masquerading as the forces of Nature (Lunar Lander) or evil aliens (Startrek).

software
software
software
software
software
software
software
software
software
software
software
software
software
software

column
column
column
column
column
column
column
column
column
column
column
column
column
column

Writing game programs can rapidly teach a large number of programming skills and techniques. When the initial fascination of writing game programs wears off, if it ever does, the computer user is often attracted to two new frontiers – applications programming and systems programming.

Applications programming is most often involved with programs designed to automate part of a small business – accounts receivable, inventory, order entry, etc. Such programs have the potential for monetary gain to the writer, and the money helps make up for the lack of glamour inherent in writing applications.

On the other hand, systems programming has an aura of mystery that attracts the newcomer. The internal workings of a BASIC interpreter, FORTRAN compiler or operating system are full of hidden secrets and obscure but elegant programming tricks. The allure of writing a systems program seems to lie in its appearance to some as a task of a "higher order" than the writing of mere applications programs.

However, both the applications programmer and the systems programmer have many responsibilities to the users of the programs they write. Programs must be documented carefully and extensively, and the programmer should be responsive to the suggestions and complaints of those who use his programs.

As the new computer user continues to delve into programming, his skills increase. Those who enter the personal computing field from a knowledge of hardware electronics soon find themselves writing software, while the programmers learn a lot about assembling and maintaining hardware systems.

The rapid growth of the number of people competent to service and program computers will have a dramatic impact on the use of inexpensive computer power. Each time the cost of computing halves, the number of potential computer installations doubles. The personal computer user finds himself at the leading edge of new computer applications and technology. He is becoming a source of expertise and innovation. He is not merely a passive, casual user of hardware and software developed by others.

Clearly, personal computers have not yet reached the *mass* market, but this is about to change with advances in technology, reductions in cost and the determination of competent manufacturers to produce home computers in quantity.

I expect the personal computer to become the kind of thing that people carry with them, a companion that takes notes, does accounting, gives reminders, handles a thousand personal tasks. Leaving everything to the computer opens the possibility of accidentally paying 20 years of insurance premiums in advance and things like that, but other kinds of accidents happen already and we learn to deal with them.

New questions are raised. In addition to playing games, calculating income taxes and all that, what are the uses of the home computer? What will inexpensive computers *do?*

These are chiefly software questions. I expect all the standard software systems to become available to the hobbyist in the course of time – APL, FORTRAN and the rest.

In this column, we'll try to talk usefully about what the personal computer user might do with the array of hardware and software tools at his disposal.

Next issue:

Using the Personal Computer for Information Retrieval

HARD TALK ABOUT

Okay, you've bought that box, the lights are blinking, but what is *really* going on? Why does a computer have two busses? What happens inside an ALU in an MPU? Do you know how to push, pull, stack, and point? If you don't know the difference between an IX, a CCR, and a PC, read this article.

By Steve Pollini

When I was first learning FORTRAN a few years ago, I had no idea how a computer computed. I had been told that a computer works with ones and zeroes, but that made no sense at all in terms of executing "DO loops" and other FORTRAN statements. This bothered me. If you have worked with a high-level language in your early programming experience, you've no doubt also wondered what was "going on behind the scenes" in your computer.

Since most home computers are microcomputers, I'll speak in terms of microcomputers. They follow people's instructions. A set of instructions to solve a particular problem or to carry out a specific function is called a *program.*

The instructions must occur in a logical order: if you enter them in a random order, the computer will spit out a random answer. Imagine, for example, a program in which you told the computer first to accept your input data; second, to output (print) the results; and third, the calculate the answers. You wouldn't get out useful data, since you told the computer to output the answers before computing them.

Easily changed programs, *software,* instruct the difficult-to-change electronic computer components, *hardware,* to do sequences of operations that produce results we want. Your effectiveness as a programmer may increase if you increase your knowledge of what happens inside the computer.

The heart of the microcomputer is the microprocessor, popularly called "the computer on a chip." It does all the processing: it does the arithmetic and controls the whole microcomputer system. It's a single flake of material, composed of thousands of transistors and other discrete devices integrated into useful circuits. Microprocessors are not only *integrated circuits, IC*'s, but are by current standards extremely large and complex circuits in the class known as *LSI*'s, Large Scale Integrated circuits.

But a microprocessor chip cannot stand alone. It must be supplied with highly regulated power. Special power supplies serve the various computer components.

Further, there must be a way for the microprocessor to get information (data) from the outside world. This calls for an *I/O (input/output) system.* This section of the computer has specialized circuitry designed to act as an interface, a liaison, between the computer and such devices as Teletypes, CRT (cathode ray tube, television-like) terminals and line printers.

For a microprocessor to function as a microcomputer, it needs a place to store programs and data. It stores them in a memory section, which has many locations called *addresses* that hold information.

All the parts of a computer – the microprocessing unit (MPU), memory and I/O – are used in running every program.

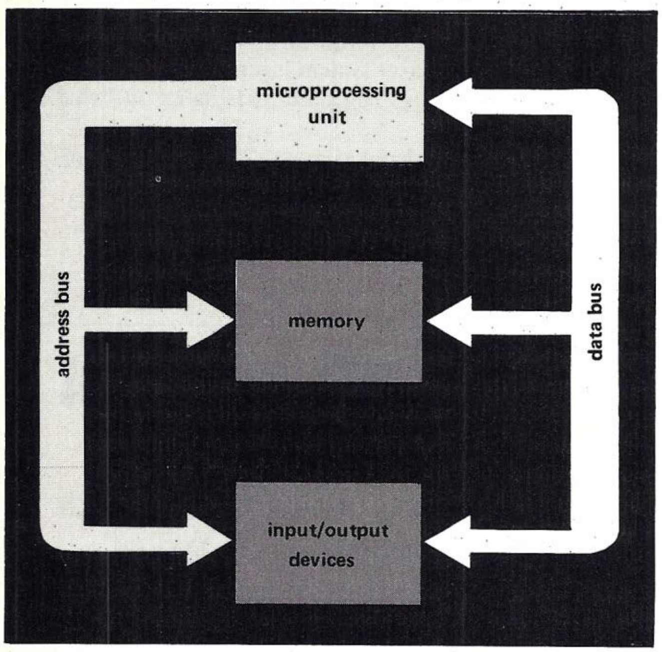
Just as a bus is used in our everyday world to transport people to and from different locations, *busses* transport information from section to section of the computer. The busses in the microcomputer are not wheeled; they transport information electrically. Electric pulses travel over wires and other electrical conductors. A bus within a computer, then, consists of groups of conductors that transport information.

The data bus consists of 8 data lines, since information is transferred and stored in 8-bit words (codes) called *bytes.* Each byte of data in the memory is assigned a unique 16-bit address. That is, each 16-bit address in the memory holds at most one byte of data. The computer goes to the correct address to drop off or pick up data. The 16-bit address code lets the computer keep track of 65,536 addresses.

The MPU can read-out-of or write-into any particular memory location. Addresses are sent from the microprocessor to the memory via the Address Bus.

Each I/O device in the system also has one or more 16-bit addresses assigned to it. These I/O devices are addressed the same way as any memory byte location. (In some microprocessors, such as the Intel 8080, there is a separate address bus, but in this article, the Motorola MC6800 will serve as

HARDWARE



A microcomputer's five parts. The microprocessing unit (MPU), memory, and I/O devices (such as Teletypes, CRT terminals, and line printers) communicate with each other via the data bus (which transfers the data) and the address bus (which says where the data should be put).

our general model.)

Let's now look inside the microprocessor.

The microprocessor contains an Arithmetic Logic Unit (ALU), which performs all of the basic arithmetic operations (add, subtract, compare, etc.). Because it must perform these operations in a particular sequence, the ALU is controlled by an MPU register called the *Program Counter (PC).* Once a program – that is, a sequence of instructions – is loaded into memory, the Program Counter is loaded with the address of the first instruction of the program. When the computer is put into the RUN mode, the MPU puts the address contained in the PC onto the address bus and reads the contents of that location via the data bus. It executes the instruction it has read, after incrementing the PC to point to the next instruction. This sequence is repeated until the processor is halted.

The MPU contains two accumulators, labeled ACCA and ACCB. They temporarily store data, either before or after the arithmetic logic unit has operated on it. For example, to add two numbers, first you must load one of the numbers from memory into ACCA. Then the other number must be loaded into ACCB. Then an add instruction (which, like the instructions to load the accumulators, will be in the program in memory) must be executed to add the contents of the accumulators. After they are added together, the result is temporarily stored in ACCA. To let you see the result, your program must have an instruction telling the MPU to store the contents of ACCA at a particular memory address location. At this memory address location could be either memory or an I/O device. (Some of you super-software-types may be grumbling now, since there is an instruction which allows data at a memory address to be directly added to ACCA. You say to first load ACCB is inefficient because it takes extra steps. I agree but, for the sake of example, chose to show the use of ACCB.)

To figure out what instruction to execute, the MPU contains an Instruction Decoder. It decodes the instructions read from the program and has the MPU perform them. Thus, when it reads an ABA (add ACCA to ACCB) in a binary form, it sets up the logic circuitry to take the contents of ACCA and ACCB, add them together in the ALU and store the result in ACCA.

The Condition Code Register (CCR) is used by the MPU to control program flow. It consists of 6 bits that can be set to either a one or a zero. A one in a particular CCR bit is considered a set condition, while a zero in a particular bit is considered a cleared condition. For example, one of the CCR bits is the Carry bit and it gets set whenever there is a carry from the most significant bit (bit 7) of a result. This could happen, for example, when adding two numbers in the accumulators.

Carry Bit		
0	1 0 1 1 1 0 1 0	ACCA before addition
0	1 1 0 0 0 1 0 1	ACCB before addition
1	0 1 1 1 1 1 1 1	ACCA after addition

Once the carry bit is set, it can be tested or checked to designate program flow, i.e., determine what part of the program

to execute next.

The Index Register (IX) is a two-byte register that temporarily stores data or a memory address. In a real-world application, you can use it to index a table. Or if you want to clear a section of memory (set all the bytes equal to zero), load a starting address into the IX. Clear the address location designated by the IX. Then increment the IX by one and clear the next byte. Continue this process until you've cleared the last address of the block. You may now ask, "How does the MPU know when the last address has been cleared?" Take care of this by comparing the IX with a memory byte containing a specified number. Make the operation end when the address in the IX equals the specified number in memory.

Finally, the MPU contains a Stack Pointer. The Stack Pointer points to the Stack. The Stack is a section of memory used for temporarily storing MPU register contents. Say, for instance, that you just finished a calculation and the answer is in ACCA. Instead of storing ACCA into a memory address location while performing another operation, it is at times more efficient to just push it onto the Stack. Using the Stack is more efficient because it takes fewer bytes to implement than a memory store and read. However, it is a sequential read-write operation rather than random access as with normal memory store. This means that each byte has to be read in a last-in-first-out basis. Random access memory usage, however, means that any byte can be accessed at any time regardless of its position.

The Push Instruction (PSHA) causes the contents of the indicated accumulator (A in this example) to be stored on the Stack, in memory, at the location indicated by the Stack Pointer. After the storage operation, the Stack Pointer is automatically decremented by one and is "pointing" to the next empty stack location.

The Pull Instruction (PULA or PULB) causes the last byte stacked to be loaded into the appropriate accumulator. Just before that data transmission, the Stack Pointer is automatically incremented by one so it will point to the last byte stacked.

The Stack is also used for linking subroutines. A *subroutine* is a program within a program. Say, for example, you have a long program that performs many multiplications. You can write a small program (subroutine) that multiplies two numbers, and go to the subroutine at every place in the program where necessary.

The Stack stores the current address of the Program Counter every time the program goes to the subroutine. This is how the MPU keeps track of where it left off in the main program. When the subroutine is finished, the Program Counter's old address is pulled off the Stack and reloaded into the Program Counter. Then the main program picks up where it left off before having called the subroutine.

In this way, you have to write the multiplication routine into the program only once. You save much memory space, which can be critical when writing long programs.

This once-over-lightly description of the inner operation of a typical microcomputer system is just enough to help the non-hardware-oriented computer user figure out what's going on in his system.

Every aspect of this deserves more discussion. We'll return to it in the future.

Portions of this article were reprinted with permission from *Computer Notes*.

a quicker way to understand your microprocessor

by Russ Walter

To get acquainted with your microprocessor, you don't have to wade through that big book the manufacturer sent you. You can learn just as much by reading a single page if it is written in Universal Assembly Language, a tool as powerful for explaining computers as algebra is for science.

UAL describes what the computer puts into "electronic boxes". (The boxes the computer accesses the most quickly are called *registers;* slower boxes are called *memory locations.*) The UAL code for putting the number 5 into box 3 is "3 ← 5". Pronounce it, "Box 3 gets 5".

Parentheses mean "the contents of". For example, 3 ← (5) means box 3 gets *the contents of* box 5. If box 5 contains 27, box 3 gets 27. Thus, 27 is in both box 3 and box 5.

The notation extends further. 3 ← ((5)) means box 3 gets the contents *of the contents* of box 5. If box 5 contains 27, and box 27 contains 42, box 3 gets 42.

If box 3 contains 46, the statement (3) ← 5 means box 46 gets 5.

UAL uses these symbols.

SYMBOL	PRONOUNCED	EXAMPLE	ANSWER	HOW TO GET THE ANSWER
+	plus	1010+1001	10011	Add, but remember binary 1+1 is 10.
–	minus	1010-1001	0001	Subtract, but remember binary 10-1 is 1.
∧	and	1010∧1001	1000	Put a 1 wherever both original numbers had 1.
∨	or	1010∨1001	1011	Put a 1 wherever at least one of the original numbers had 1.
⊻	exclusive or	1010⊻1001	0011	Put a 1 wherever *just* one of the original numbers had 1.
~	not	~10011011	01100100	Replace 1 by 0, and 0 by 1.
0→	logical shift right	0→ 10011011	01001101	Write 0, then the original number omitting the rightmost bit.
x→	arithmetic shift right	x→ 10011011	11001101	Write the leftmost bit, then the original number omitting the rightmost bit.
⊂→	rotate right	⊂→ 10011011	11001101	Write the rightmost bit, then the original number omitting the rightmost bit.

A large IBM 370 can do the following, but microprocessors cannot because most can add and subtract only binary integers:

UAL SYMBOL	OPERATION
*	multiply
/	divide
⊕	add real numbers (numbers containing a few digits after the decimal point)
(⊕)	add double precision numbers (numbers containing many digits after the decimal point)
⊞	add in base ten (instead of in binary)

Every computer has a Program Counter, which UAL calls the PC. While the computer is executing an instruction, the PC contains the location of the next instruction. For example, while the computer executes an instruction in box 52, the PC contains the number 53.

Some instructions are long-winded enough to occupy two or even three boxes. While the computer executes an instruction occupying boxes 71, 72 and 73, the PC contains the number 74.

Every computer has a flag register, which UAL calls F. Whenever the computer does arithmetic, the flag register's bits change to reflect whether the answer was positive, negative, zero, needed a carry into the next word or *overflowed* (was too large or too negative for the computer to handle).

In UAL if you say "Test -5 + 2", the computer adds -5 to 2 and changes F appropriately but doesn't put the answer -3 in any box.

Most computers have a Stack Pointer, which UAL calls the SP. It is a register that keeps track of subroutines.

Now you know the essentials of UAL. Let's use it to analyze the two most popular microprocessors – Motorola's M6800 (used in the Altair 680) and Intel's 8080 (used in the Imsai 8080 and the Altair 8800).

Motorola's M6800 contains a 16-bit Program Counter (PC), a 16-bit Stack Pointer (SP), an 8-bit Flag register (F) and three other 8-bit registers called A, B and X. The two leftmost bits of F are both 1; the other bits are called H (Half-carry), I (Interrupt mask), N (Negative), Z (Zero), V (oVerflow) and C (Carry). Each memory location holds 8 bits.

When the computer is processing a list of information, the X register keeps track of where in the list the computer is. For example, if the computer is processing the 43rd item, the X register contains the number 43, which is called the

table 1 – M6800 instruction set

M6800 assembler	UAL	M6800 machine language	µsec
No OPeration	no operation	01	2
Transfer A to Program status	F ← (A)	06	
Transfer Program status to A	A ← (F)	07	
INcrement X	X ← (X)+1	08	4
DEcrement X	X ← (X)−1	09	
CLear V	V ← 0	0A	2
SEt V	V ← 1	0B	
CLear C	C ← 0	0C	
SEt C	C ← 1	0D	
CLear I	I ← 0	0E	
SEt I	I ← 1	0F	
SuBtract Accumulators	A ← (A)−(B)	10	
Compare B with A	test (A)−(B)	11	
Transfer A to B	B ← (A)	16	
Transfer B to A	A ← (B)	17	
Decimal Adjust A	if (A_{right})>9, A←(A)+6 if (A_{left})>9, A_{left}←(A_{left})+6	19	
Add B to A	A←(A)+(B)	1B	
BRanch Always ab	PC←(PC)+ab	20ab	4
Branch if HIgher ab	if (C)v(Z)=0, PC←(PC)+ab	22ab	
Branch if C Clear ab	if (C)=0, PC←(PC)+ab	24ab	
Branch if Not Equal ab	if (Z)=0, PC←(PC)+ab	26ab	
Branch if V Clear ab	if (V)=0, PC←(PC)+ab	28ab	
Branch if PLus ab	if (N)=0, PC←(PC)+ab	2Aab	
Branch if Greater or Equal ab	if (N)v(V)=0, PC←(PC)+ab	2Cab	
Branch if Greater Than ab	if (Z)v[(N)v(V)]=0, PC←(PC)+ab	2Eab	
Transfer Sp to X	X←(SP)+1	30	
INcrement Sp	SP←(SP)+1	31	
PULl A	SP←(SP)+1 A←((SP))	32	
PULl B	SP←(SP)+1 B←((SP))	33	
DEcrement Sp	SP←(SP)−1	34	
Transfer X to Sp	SP←(X)−1	35	
PuSH A	SP←(A) SP←(SP)−1	36	
PuSH B	SP←(B) SP←(SP)−1	37	
ReTurn from Subroutine	PC←(1+(SP)) (2+(SP)) SP←(SP)+2	39	5
ReTurn from Interrupt	F←(1+(SP)) B←(2+(SP)) A←(3+(SP)) X←(4+(SP)) (5+(SP)) PC←(6+(SP)) (7+(SP)) SP←(SP)+7	3B	10
WAit for Interrupt	(SP)−1 (SP)←(PC) (SP)−3 (SP)−2←(X) (SP)−4←(A) (SP)−5←(B) (SP)−6←(F) SP←(SP)−7	3E	9
SoftWare Interrupt	(SP)−1 (SP)←(PC) (SP)−3 (SP)−2←(X) (SP)−4←(A) (SP)−5←(B) (SP)−6←(F) SP←(SP)−7 I←1 PC←(h−5) (h−4)	3F	12
NEGative A	A←−(A)	40	2
COMplement A	A←~(A)	43	
Logical Shift Right A	A←(A) (C)	44	
ROtate Right A	A← (A) (C)	46	
Arithmetic Shift Right B	A←(A) (C)	47	
Arithmetic Shift Left A	A←(C) (A)	48	
ROtate Left A	A←(C)(A)	49	

M6800 assembler	UAL	M6800 machine language	µsec
DECrement A	A←(A)−1	4A	2
INCrement A	A←(A)+1	4C	
TeST A	test (A)	4D	
CLeaR A	A←0	4F	
JuMP ab, X	PC←ab+(X)	6Eab	4
JuMP abcd	PC←abcd	7Eabcd	3
SuBtract from A #ab	A←(A)−ab	80ab	2
CoMPare A #ab	test (A)−ab	81ab	
SuBtr with Carry fr A	A←(A)−ab−(C)	82ab	
AND A #ab	A←(A)∧ab	84ab	
BIT A #ab	test (A)∧ab	85ab	
LoaD Accumulator A #ab	A←ab	86ab	
Exclusive OR A #ab	A←(A)⊻ab	88ab	
ADd with Carry to A #ab	A←(A)+ab+(C)	89ab	
OR Accumulator A #ab	A←(A)vab	8Aab	
ADD A#ab	A←(A)+ab	8Bab	
ComPare X #ab	test (X_{left})−ab and (X_{right})−cd	8Cab	3
Branch to SubRoutine ab	(SP)−1 (SP)←(PC) SP←(SP)−2 PC←(PC)+ab	8Dab	8
LoaD S abcd	SP←abcd	8Eabcd	3
STore Accumulator A ab	ab←(A)	97ab	4
STore S	ab ab+1←(SP)	9Fab	5
Jump to SubRoutine ab, X	(SP)−1 (SP)←(PC) SP←(SP)−1 PC←ab+(X)	ADab	8

Table 1 Here's what the second line of the table means . . . In your assembly-language program, you can say TAP. It stands for "Transfer A to Psw". It means "F gets the contents of A". The assembler will translate the letters TAP into the hexadecimal code 06 (actually the binary code 00000110). When you tell the computer to execute the program, the 00000110 makes F get the contents of A in 2 microseconds.

Line 86ab says, for example, that "LDAA #35" stands for "LoaD Accumulator A #35", which means "register A gets 35". The assembler will translate LDAA #35 into the hexadecimal code 8635 (actually the binary code 1000011000110101). When you tell the computer to execute the program, the 1000011000 110101 makes register A get 35 in 2 microseconds.

In line 3F, "h" means the highest location in your memory.

To send information to a peripheral device (such as a Teletype), move the information to the device's memory location.

You can derive the missing lines from these rules:

In lines 22ab, 24ab, 26ab, etc., replacing 0 by 1 gives lines 23ab, 25ab, 27ab, etc.

In lines 40 thru 4F, replacing A by B gives lines 60ab thru 6Fab, which take 7 microseconds. Replacing A by abcd gives lines 70abcd thru 7Fabcd, which take 6 microseconds.

In lines 80ab thru 8Eabce, replacing ab by (ab) and replacing cd by (ab+1) give lines 90ab thru 9Eabcd, which take 1 more microsecond.

In lines 90ab thru 9Fab, replacing ab by ab+(X) gives lines A0ab thru AFab, which take 2 more microseconds. Replacing ab by abcd gives lines B0ab thru BFab, which take 1 more microsecond.

In lines 80ab thru BFab, replacing A by B and replacing SP by X give lines C0ab thru FFab.

Table 2 In lines 20r thru 27r, replacing "r" and "(r)" by "a" and replacing the last capital letter in the assembler column by "I" gives lines 306a thru 376a, which take 3.5 microseconds. For example, line 306a says "ADI a" means "A ← (A)+a".

The timing (in microsceonds) is for the original 8080A. Multiply by .65 for the 8080A-1, but by .75 for the 8080A-2.

To send information to a peripheral device, move the information to the device's port.

table 2 – 8080 instruction set

8080 assembler	UAL	8080 machine language					microseconds
		2 bits	3 bits	3 bits	8 bits	8 bits	
No OPeration	no operation	0	0	0			2
Load X immediate x, ab	x←ab	0	x	1	a	b	5
Double ADd x	HL←(HL)+(x)	0	x+1	1			
STore A in X B	(BC)←(A)	0	0	2			3.5
LoaD A from X B	A←((BC))	0	1	2			
STore A in X D	(DE)←(A)	0	2	2			
LoaD A from X D	A←((DE))	0	3	2			
Store HL Direct ab	ab+1 ab←(HL)	0	4	2	a	b	8
Load HL Direct ab	HL←(ab+1)(ab)	0	5	2	a	b	
STore A ab	ab←(A)	0	6	2	a	b	6.5
LoaD A ab	A←(ab)	0	7	2	a	b	
INcrement X x	x←(x)+1	0	x	3			2.5
DeCrement X x	x←(x)−1	0	x+1	3			
INcrement R r	r←(r)+1	0	r	4			2.5 (5 if M is involved)
DeCrement R r	r←(r)−1	0	r	5			
MoVe Immediate r,a	r←a	0	r	6	a		3.5 (5 if M is inv)
Rotate A Left	A←(Carry)(A)	0	0	7			2
Rotate A Right	A←(Carry)(A)	0	1	7			
Rotate Left skip Carry	A←(A)	0	3	7			
Rotate Right skip Carry	A←(A)	0	3	7			
Decimal Adjust A	if $(A_{right})>9$, A←(A)+6 if $(A_{left})>9$, $A_{left}←(A_{left})+6$	0	4	7			
CoMplement A	A←~(A)	0	5	7			
SeT Carry	Carry←1	0	6	7			
CoMplement Carry	Carry←~(Carry)	0	7	7			
HaLT	halt	1	6	6			halt
MOVe r_1, r_2	$r_1←(r_2)$	1	r_1	r_2			2.5 (3.5 if M is inv)
ADD r	A←(A)+(r)	2	0	r			2 (3.5 if M is involved)
ADd Carry r	A←(A)+(r)+(Carry)	2	1	r			
SUBtract r	A←(A)−(r)	2	2	r			
SuBtract Borrow r	A←(A)−(r)−(Carry)	2	3	r			
ANd A r	A←(A) ∧ (r)	2	4	r			
eXclusive OR A r	A←(A) $\veebar$ (r)	2	5	r			
OR A r	A←(A)V(r)	2	6	r			
CoMPare r	test (A)−(r)	2	7	r			
Return if c	if c, PC←((SP)+1)((SP)) SP←(SP)+2	3	c	0			2.5 or 5.5
RETurn	PC←((SP)+1) ((SP)) SP←(SP)+2	3	1	1			5
POP p	p←((SP)+1) ((SP)) SP←(SP)+2	3	p	1			
PC HL	PC←(HL)	3	5	1			2.5
SP HL	SP←(HL)	3	7	1			
Jump if c ab	if c, PC←ab	3	c	2	a	b	5
JuMP ab	PC←ab	3	0	3	a	b	
OUT a	port a←(A)	3	2	3	a		
IN a	A←(port a)	3	3	3	a		
eXchange Top HL	exchange HL with (SP)+1 (SP)	3	4	3			9
eXCHanGe	exchange HL with DE	3	5	3			2
Disable Interrupts	disable interrupts	3	6	3			
Enable Interrupts	enable interrupts	3	7	3			
Call if c ab	if c, (SP)−1 (SP)−2←(PC) PC←ab SP←(SP)−2	3	c	4	a	b	5.5 or 8.5
CALL ab	(SP)−1 (SP)−2←(PC) PC←ab SP←(SP)−2	3	1	5			8.5
PUSH p	(SP)−1 (SP)−2←(p) SP←(SP)−2	3	p	5			5.5
ReSTart a	(SP)−1 (SP)−2←(PC) PC←8∗a SP←(SP)−2	3	a	7			

index. Since the X register's main purpose is to hold an index, it is called the *index register.* The A and B registers are called *general-purpose registers* or *accumulators.*

To save space, I'll write commands in hexadecimal (hex) instead of in binary. This table will help you convert between number systems.

DECIMAL	BINARY	HEX	DECIMAL	BINARY	HEX
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

The complete list of commands the M6800 understands is in Table 1.

Intel's 8080 contains a 16-bit PC, 16-bit SP, 8-bit F and seven other 8-bit registers called A, B, C, D, E, H and L. Of the 8-bit registers, the A register is used the most heavily and therefore called the *accumulator.* The most important bits of F are called Carry, Zero, Negative and *Even Parity* (which checks whether the quantity of ones in the answer is even). Each memory location holds 8 bits.

The computer has fake 16-bit registers. The "BC register" is actually the 8 bits of the B register followed by the 8 bits of the C register. The "DE register" is the 8 bits of D followed by the 8 bits of E. The "HL register" is the 8 bits of H followed by the 8 bits of L. The "PSW register" (Program Status Word) is the 8 bits of F followed by the 8 bits of A.

M means (HL). For example, if (HL) is 327 and you tell the computer to put 5 into the "M register", it will put 5 into Memory location 327.

The registers and conditions are numbered:

Registers R	Pairs P	Index Pairs X	Conditions C
0 B	0 BC	0 BC	0 Zero
1 C			1 Not Zero
2 D	2 DE	2 DE	2 Carry
3 E			3 Not Carry
4 H	4 HL	4 HL	4 sign Plus
5 L			5 sign Minus
6 M	6 PSW	6 SP	6 Parity Even
7 A			7 Parity Odd

The complete list of commands the 8080 understands is in Table 2.

Which is better, the M6800 or the 8080? The 8080 has the advantage of more registers, 16-bit arithmetic and an exchange command. On the other hand, it is slower, cannot detect overflow, cannot do logical and arithmetic shifts, lacks a true index register, requires a full 3 bytes to change the PC and insists that all 8-bit operations – even input and output – use the A register.

Zilog's Z-80 is an improvement over the 8080. The Z-80 detects overflow, does arithmetic and logical shifts, has a true index register, requires only 2 bytes to change the PC and provides alternatives to the A register. To accomplish those feats, the Z-80 includes extra registers and op codes. The only 8080 weakness the Z-80 doesn't overcome is the slow speed: the Z-80 cycles at the same speed as the 8080A.

A program written for the 8080 will also work on the Z-80, if the program doesn't rely on the parity bit (which the Z-80 sometimes treats as an overflow bit). MITS BASIC uses the parity bit and will *not* run on a Z-80.

Some programmers think the Z-80 will become the most popular microprocessor. I'll take a detailed look at it in the next issue.

MICRO PROCESSOR SURVEY

FELLOW COMPUTERER . . .

WOULD YOU TAKE A COUPLA MINUTES AND LET ME KNOW THE EXTENT YOU . . . AND YOUR COMPANY . . . ARE USING MICROPROCESSORS.

THESE ARE THE COMPUTERS ON A CHIP, OR MATCHBOOK COMPUTERS, AS THE CALIFORNIANS CALL THEM.

LIKE THE MINICOMPUTERS OF A FEW YEARS AGO, THESE LITTLE GADGETS WILL CAUSE ANOTHER REVOLUTION, TOO. LOOK WHAT THEY HAVE DONE TO THE PRICE OF HAND CALCULATORS.

TO GEORGE AHMUTY, 606 WENDY LANE, WESTPORT, CT 06880

DEAR GEORGE,

MY COMPANY USES . . DOES NOT USE . . MICROPROCESSORS.

I HAVE BUILT A MICROCOMPUTER . . . HAVE NOT . . . IF YES, WHICH KIND . . . .

I AM INTERESTED IN JOINING A SPECIAL INTEREST GROUP TO LEARN ABOUT THESE MACHINES.

MY NAME . . . .

MY HOME ADDRESS . . . .

MY CITY STATE ZIP . . . .

PHONE NO . . . .

GEORGE, GET AHOLD OF THESE PROSPECTIVE MICROISTS, TOO:

CONNECTICUT MICROISTS, 60111 WENDY LANE, WESTPORT, CT 06880 (203) 227-8534

FUTURE COMPUTING

Dick Heiser aspires to be the Grand Old Man of Computer Retailing. He has a good start. When he and his wife, Lois, opened the Computer Store in Santa Monica, Calif. in October 1975, they were pioneers who were venturing into the unknown.

They chose to operate a high-class specialty shop, uncluttered, staffed with extremely knowledgeable people whose interest lies in helping customers and exuding a sense of competence in dealing with the systems of the future. (Note that some other operators have chosen the surplus-store-with-cheap-computers approach just as deliberately and with appreciable success. Styles do vary.)

Even with the good start, Dick has his work cut out for him to maneuver his new enterprise through the decades without a disaster. He wants to be a respected and well-to-do Grand Old Man. His success will depend on his ability to read the future of his field correctly, anticipate trends, influence them as best he can, and bet cash and credit on his judgment day after day.

The opinions of a fellow with these credentials are bound to be interesting, so PERSONAL COMPUTING *pressed him for an analysis of the fairly near-term future of the market in which he hopes to prosper. His carefully considered prophecy came through in the form of a tight list of terse statements.*

Others may disagree, but here's the Heiser view of personal computing hardware as it will appear in the stores two years from now, say in the first quarter of 1979.

Central Processor

Standard will be an 8-bit MOS single-chip processor. Zilog Z-80 chip will be most popular.

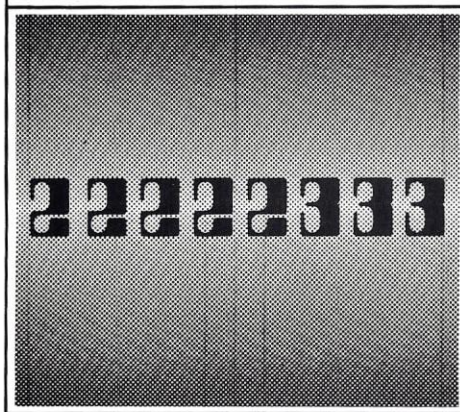
Some machines, especially "number crunchers" (for statistical analysis and numerical simulations) will use 16-bit processors like the TI9900 or MOS Technology's counterpart.

Memory will be MOS RAM, median size 16K bytes, with a range of 8K to 64K bytes. The memory card will hold 32K bytes. The first 1K of memory will always be ROM.

CPU will communicate with memory over a short, wide bus with provision for 24-bit memory addresses, but most machines will be sold without the memory-mapping option and without the memory bus extender and so will use only the low-order 16 bits for memory addressing.

CPU will communicate with I/O controllers over a longer, but narrower bus. Some manufacturers are already moving in that direction.

There will be a clock readable and settable under program control for timing events and displaying time-of-day.



Time will be measured as HH:MM:SS:mmm.

Price of the CPU, bus and 16K memory will be $700.

Mass Storage

Most home units will feature digital cassette units with motion control electronics.

Digital cassette drive will operate like a DECtape, with fixed blocks of data so that data can be randomly accessed by block number. Data can be rewritten in-place.

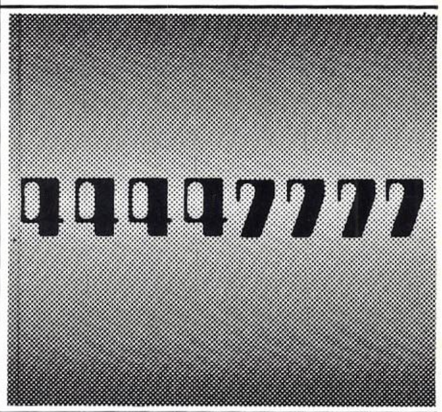
Density will be about 800 BPI with parity or cyclic error detection.

Cassette drive will have a fast search, so that any block of data can be accessed in 20 seconds.

Read-after-write would be nice, but it's the kind of feature that's not widely understood. Many such "unnecessary" features will be on the 1979 systems, however, because the consumer will have a high expectation for the reliability and fidelity of his home computer. The digital cassette will cost about $250.

Disks will be available as an option, with a dual floppy on a single spindle offering 2 megabytes of on-line storage. About $2000.

Ordinary audio tapes will be used for loading some old systems, but most people won't understand the operating procedures, and few dealers will be willing to carry a product that requires so much training and assistance to the customer.



Keyboard

Really high quality! This is "where the rubber meets the road" in winning customer acceptance.

Upper case/lower case. All programs will accept either upper case or mixed upper/lower case. Text edit-

Continued

ing will be so important an application that any computer without lower case will be regarded as "special purpose."

Layout will be either standard typewriter-style or 4-row alphabetic (abcdef. . .).

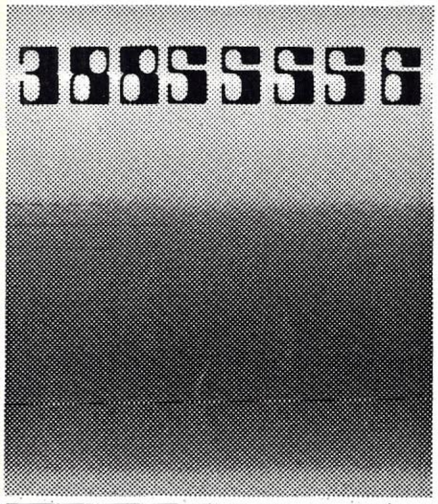
Large-format function keyboards will be available for super-Startrek consoles, special application programs, etc. They will use either plastic "swiss-cheese" masks over pushbuttons or plastic overlays over touchpanels.

Price for the alphanumeric keyboard will be $150.

Video Display

Black/white, upper/lower case, 16 lines of 64 characters.

FCC approval for direct RF hookup to TV would be extremely favorable to mass acceptance of many kinds of home games and electronics.



Introduction of true vector graphics controllers will offer exciting displays of complex drawings and games. This will be optional and will be separate from the alpha video controller, but will probably supersede the bit-map display.

Retail price of the video display would be about $150, but most systems will have the video (and the $150) integrated into the CPU.

Video Monitor

High-resolution 9" monitors will cost $200 by late 1978.

A 15" monitor will cost $325 by then (black and white).

Printer

This whole matter is speculative. Something is badly needed, but efforts to date have offered less success than in any other area.

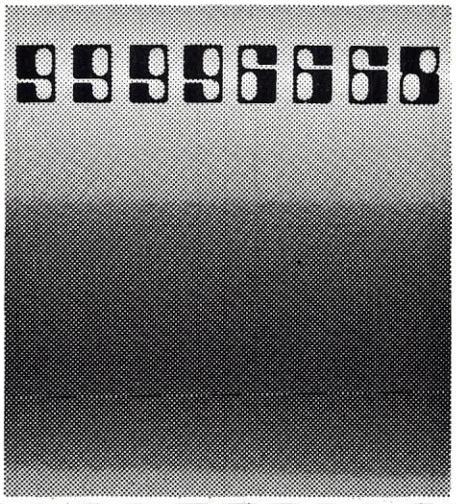
Low-cost Selectric bases should offer the right kind of economy, but mass acceptance of this sort of product will depend on a costly assembly and reconditioning service to provide units that are interfaced and ready-to-run.

Matrix printing, upper/lower case, 80-column printers will provide higher speed for more money.

Reconditioned Selectrics with RS-232 baseplates could be delivered for about $1000.

Missing Features

Interrupts won't be needed, since a home computer is inherently a single-user system. Maybe some vendor will add interrupts as a flashy feature, but they'll be impractical for most purposes.



Power fail detection and recovery won't be needed.

The front panel needs only a power switch and a reset switch. Any extra switches, knobs, lights and displays will have to be financed by increased sales rather than by system requirements.

Multiprogramming and storage protection. Not needed.

Options

Mass Storage
- Second tape unit
- Floppy disk
- Rigid disk
- CCD memory expansion

Input/Output
- Modem and coupler (for communication with other systems and databases; very little for "time-sharing")
- Motion graphics (A/D to run bit-map displays)
- Joysticks
- Audio output

Opportunities for Improvement

Color graphics (too complicated to provide adequate software, too difficult for home computerists to use).

Parity checking of internal memory and external storage would provide reliability that would enable a company to develop an outstanding reputation with long-range sales benefits.

System Budget

CPU with 16K memory	$ 700
Tape unit	250
Keyboard	150
Video controller	150
Monitor	200
Selectric printer	1000
Total	$2450

Overall Comments

1979 computer will be a quality product. All makes will perform surprisingly well. Just as we now expect a hand calculator to work every time, we'll expect the computers to work every time.

Assembled units will dominate the marketplace. They'll have been burned-in for at least 72 hours and should require no more service than a television set. Service contracts will be available.

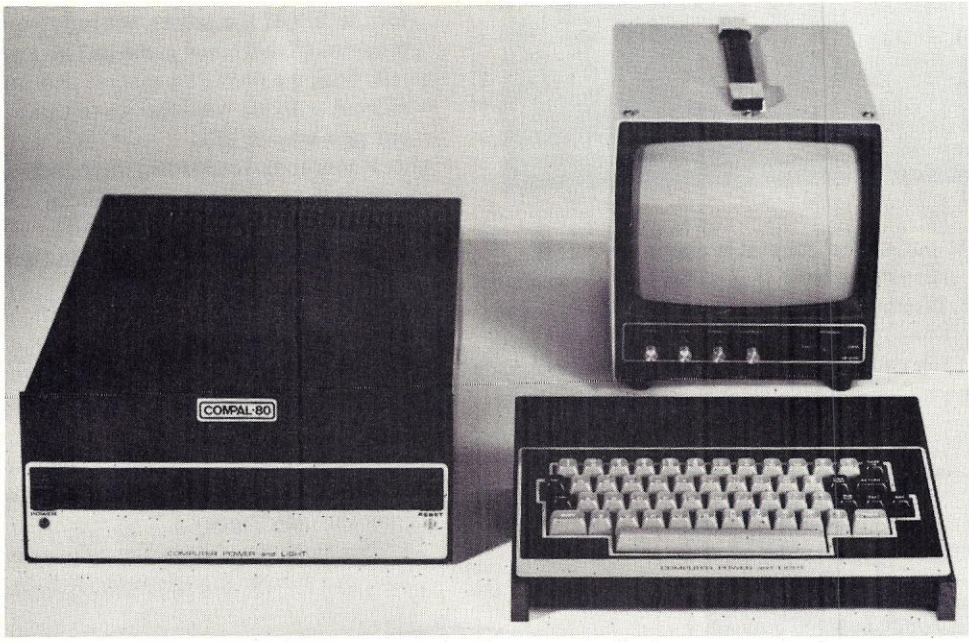
Computers will be sold almost entirely through dealers. Various dealers will present distinctively different products and services. The primary types will be: volume dealers, custom hardware dealers, hardware/software dealers and very small shops.

In making these crisp remarks on future computing, Dick Heiser was torn between mid-1978 and early-1979 as the critical times on where to place these predictions. He decided to go with 1979 on the theory that one always tends to overestimate progress in the short run and underestimate it in the long run. This is a short-run-prediction and he chose to be a little more conservative.

Software is another subject for discussion and Heiser was about to address it when duty called. Somebody *had to mind the store.* PERSONAL COMPUTING *will press him for a list of his anticipations in software sales at retail.*

Other prophets are being solicited for their visions of the future, both near and distant.

SEE THE LIGHT



The COMPAL-80 computer system: $1863.00

WHAT YOU SEE IS WHAT YOU GET!
The computer, with 12K memory, input/output ports, keyboard, 9" video monitor, all enclosures, and powerful extended BASIC language – Just plug it in and you're ready to go!

WE'RE READY NOW.
The COMPAL-80 is a complete, fully-assembled guaranteed computer that is available now. Put it to work for you on home budgets, checkbook balancing, investment profiles, mailing lists, record/book/recipe indexes, text editing, mathematical/logical skill development, word games, strategy games, sports simulations, and more . . . We have ready-made programs available,

or if you'd rather,
WE'LL SHOW YOU HOW
to write your own programs!
Anyone can write and speak eloquently in computer language after attending our series of classes which deal directly and efficiently with the art of programming.

THE COMPAL-80 IS FULLY EXPANDABLE.
You can easily increase the power of the COMPAL-80 computer, by adding more memory, a printer, plotter, dual floppy disk drive, etc. . . .
Inquire about our business systems with dual floppy disk and printer, in a fully-coordinated system including disk BASIC and a variety of business software packages.

COMPUTER™
POWER & LIGHT INC.

12321 Ventura Blvd., Studio City, CA 91604 (West of Laurel Canyon) 213 760-0405
Open 12 to 10 Tuesday-Thursday, 12 to 5 Friday-Sunday (closed Mondays)

Bank financing arranged. Also Credit Cards and Cash.

CIRCLE 27

GAMES & EXERCISES

Although some purists feel miserable about the practice, most newcomers to computing these days seem to begin by playing games. Literally, playing games. The newcomer responds to questions typed out by the machine on paper or displayed on a television terminal. The illusion of power, or at least of some influence, is so great that the game player is drawn deeper and deeper into the interactive process.

A few novices press on curiously to learn more about the systems, discipline themselves sternly, acquire the skills and knowledge to exercise *real* influence over computers and design hardware and software to accomplish new tasks well.

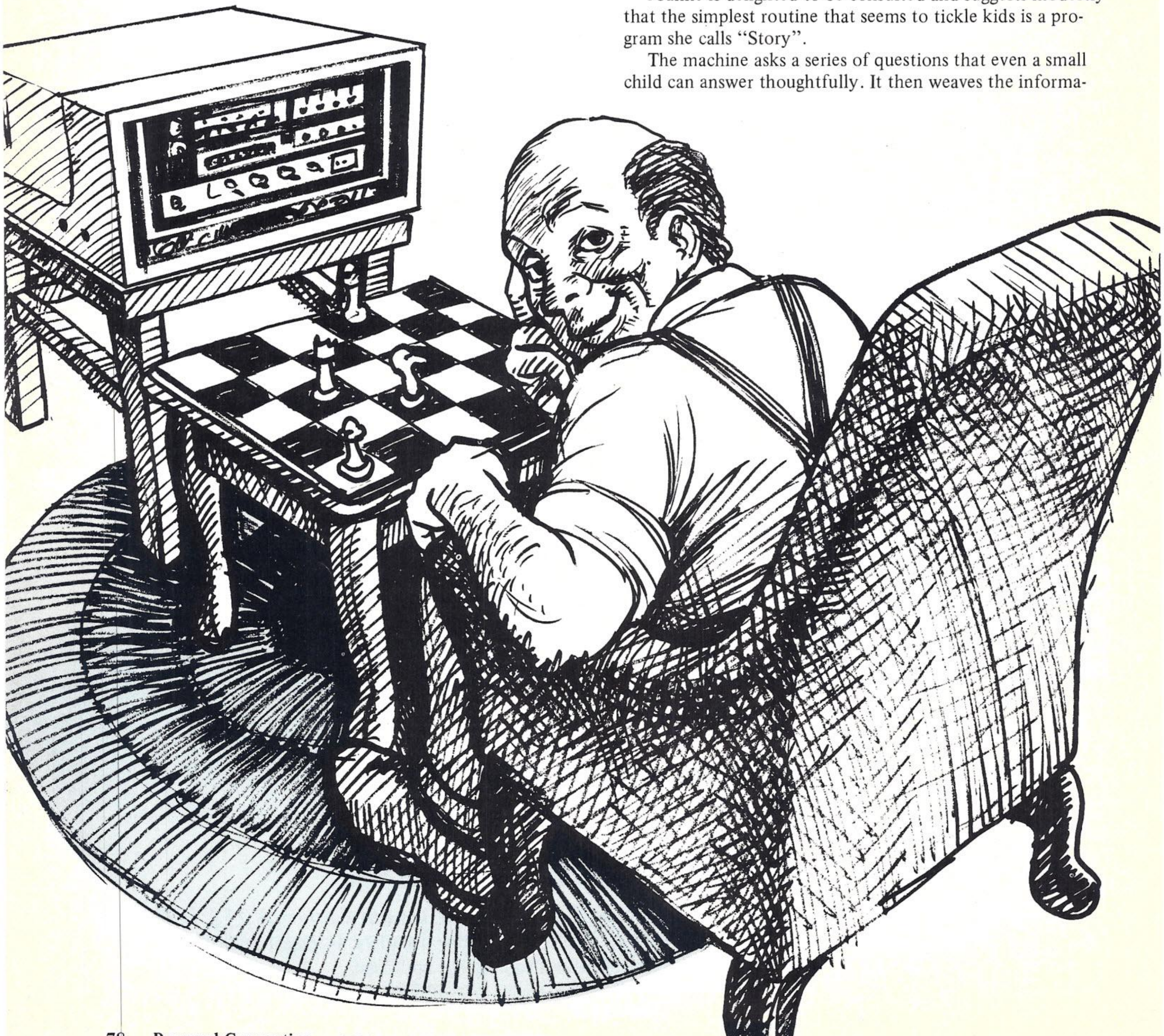
Most of us (and it is we who so annoy the purists) settle for the fun of the games, venturing only slightly further to program a trivial new game or two and retire with our laurels. Still, we provide a market and support for the more diligent workers who choose to make computers and computing a way of life.

If you look forward to introducing a young child to the computer on which you have been doing your bookkeeping (and assuming that your computer will accept BASIC in some form), you may want to find a couple of simple games that are attractive and reassuring to your uneasy young subject.

Odds are good that you'll ask Dragon Bob Albrecht for recommendations. He will steer you to Joanne Verplank (call in the afternoon) of the People's Computing Center in Menlo Park, Calif.

Joanne is delighted to be consulted and suggests modestly that the simplest routine that seems to tickle kids is a program she calls "Story".

The machine asks a series of questions that even a small child can answer thoughtfully. It then weaves the informa-



```
GAME? STORY

***STORY***


WHAT IS YO UR NAME? JOANNE
WHO DO YOU LO VE? MY MOM
WHAT ARE YO U AFRAID OF? BEARS
WHAT IS YOUR FAVORITE COLOR? PINK
WHO IS YOUR BEST FRIEND? WARREN

THANKS! HERE IS YOUR STORY


ONCE UPON A TIME JOANNE AND WARREN
WENT TO THE MO UNTAINS ON A PINK HORSE
NOBODY COULD BELIEVE IT. THEY PARKED THE HORSE
AND FOUND A SHACK. THEN THEY HEARD THE SOUND OF BEARS
BEARS ALMO ST ATE THEM, BUT MY MOM CAME AND SCARED
BEARS AWAY.  THIS IS THE END.  GOODBYE.
```

```
5000    !START STORY HERE
5005    &:&:&
5010    INPUT"WHAT IS YOUR NAME";N$
5020    INPUT"WHO DO YOU LOVE";L$
5030    INPUT"WHAT ARE YOU AFRAID OF";A$
5040    INPUT"WHAT IS YOUR FAVORITE COLOR";C$
5050    INPUT"WHO IS YOUR BEST FRIEND";F$
5060    &:&:&"THANKS! HERE'S YOUR STORY":&:&:&:&
5065    RANDOMIZE
5070    V=INT(3*RND(0)+1)
5071    L=INT(3*RND(0)+1)
5072    T=INT(3*RND(0)+1)
5075    V$(1)="HORSE":V$(2)="DINOSAUR":V$(3)="PIG"
5080    L$(1)="BEACH":L$(2)="MOUNTAINS":L$(3)="DESERT"
5085    T$(1)="CAVE":T$(2)="HOLE":T$(3)="SHACK"
5090    &"ONCE UPON A TIME "N$" AND "F$
5091    &"WENT TO THE "L$(L)" ON A "C$" ";V$(V)
5092    &"NOBODY COULD BELIEVE IT. THEY PARKED THE "V$(V)
5093    &"AND FOUND A "T$(T)". THEN THEY HEARD THE SOUND OF "A$
5094    &A$" ALMOST ATE THEM BUT "L$" CAME AND SCARED"
5095    &A$" AWAY.  THIS IS THE END. GOODBYE."
5096    &:&:&:&:&:&:&:&:&
5097    GO TO 5000
5098    END
```

tion into a personal story designed for child-catching.

The top of this page shows a conversation between the machine and a child. Also you see the listing (lines 5000-5098) Joanne prepared and said we could publish for the first time.

Joanne also provided a sample game and listing for the game called "Hamurabi", in which the player is invited to try his hand at governing ancient Sumeria. The game is more complex by far than "Story", less complex by far than "StarTrek." Not a bad practice piece, and it comes with some interesting history.

Joanne suggested that we ask David Ahl for permission to print the game. Ahl did give the idea his personal blessing with the caveat that we call attention to the book *101 BASIC Computer Games,* which is available from *Creative Computing* at $7.50. Hamurabi is among the vaunted 101, though in a slightly different version.

Ahl points out he didn't originate Hamurabi but is one of many people who have updated the original and modified it for various computer systems. We're publishing only one of the versions.

Hamurabi was designed not merely for play but for education of the young. When Computer Aided Instruction was pumped up strongly a few years ago by the expenditure of Federal education funds, educational program material was created to feed the computers that appeared suddenly in

GAMES

HAMURABI

schools all over the country. Many games were developed, some dreary and some delightful, though a loud war raged around the concept that kids could get anything beneficial out of "playing games with a computer."

The war has not been settled. PERSONAL COMPUTING will not attempt to settle it.

However, some first-hand experience suggests that the games in general (including Hamurabi specifically) have been useful in at least one case.

For several years, the Santo Domingo Indian School in New Mexico had a CAI installation built around a Hewlett-Packard 2000 computer system. The computer and a couple of dozen Teletype terminals were in a special small building just outside the school on the wide-open mesa.

The students, many speaking Pueblo languages better than English, traditionally are unenthusiastic scholars. Their problems in coping with conventional American schools are not trivial. Unclear was whether the computer system would be welcome and feasible in a school where perhaps 30% of the kids don't attend during hunting season. (The boys hunt; the girls cook. The mere presence of a foreign school does not quickly alter several thousand years of tradition.)

In practice, the Indian children run from the school to the computer room. Chattering in the language of the Pueblo, which is completely unintelligible to all the teachers, the kids fling themselves upon the terminals and set to work at math and spelling drills, eager to finish so they can play the many games. Second-graders not only puzzle out the mysteries presented by the terminals but can be seen occasionally demonstrating the system and solemnly teaching their parents and grandparents the art of playing games with the most sophisticated technology our society offers.

The Indian grade-schoolers who play Hamurabi are exposed to more than puzzles; they are exposed to archaic language and to some concepts on the nature of societies. Students who go on to high school are less likely to be buffaloed by Dickens, by "social studies" and by the trappings of outside society. Their computer has given them experience and information of value. (Ahl remarks that some educators have gone wild over the sociological implications of Hamurabi and have laid more emphasis on them than they deserve. More elaborate versions of the same game, "King," for example, have been devised to suit the sociologists.)

The H-P system is gone from Santo Domingo School now, taken off to the headquarters of the All Indian Pueblo Council to be used more efficiently in serving larger numbers of students.

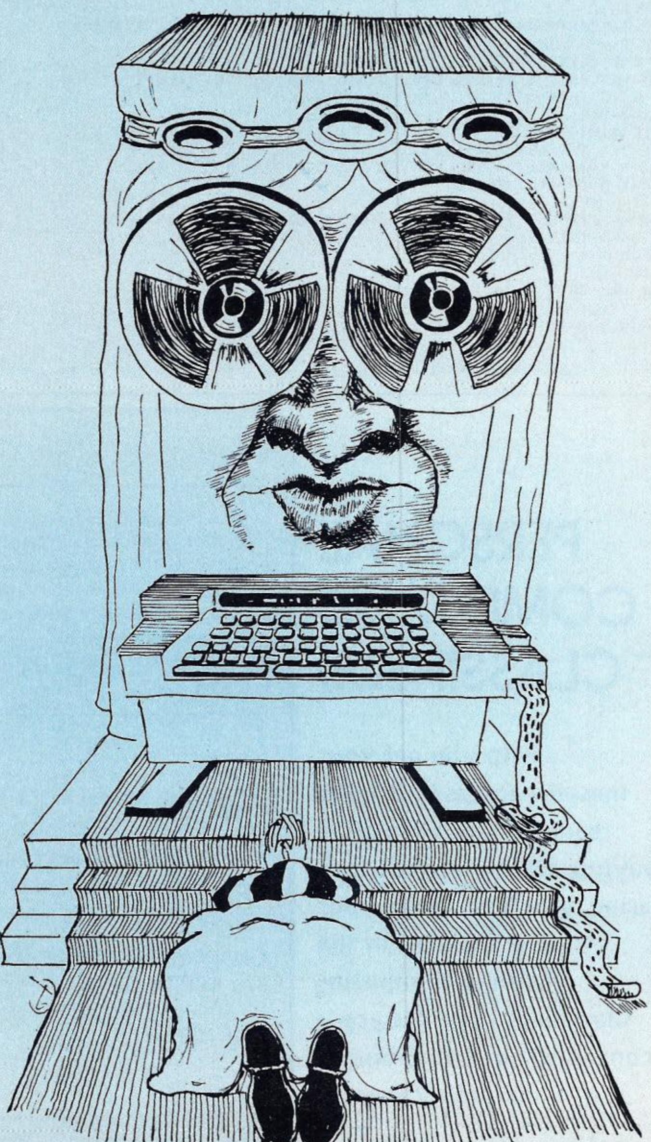
The All Indian Pueblo Council has purchased an Altair 8800A with 16 I/O ports and a pair of floppy disk units for smaller groups of kids. The hobbyist's computers are moving to the pueblos. Hamurabi returns.

On the right you see a sample Hamurabi game. Use Hamurabi to trap your novice subjects into the world of computers.

The program listing is on page 82. No one has explained whether "Charlemange" represents a typographical error or an editorial comment on a Frankish king.

```
RUN
HMRABI  07:09 PM        14-SEP-76
TRY YØUR HAND AT GØVERNING ANCIENT SUMERIA
SUCCESSFULLY FØR A 10-YR TERM ØF ØFFICE.


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 1 , 0 PEØPLE STARVED, 5 CAME TØ THE CITY.
PØPULATIØN IS NØW 100
THE CITY NØW ØWNS 1000 ACRES.
YØU HARVESTED 3 BUSHELS PER ACRE.
RATS ATE 200 BUSHELS.
YØU NØW HAVE 2800 BUSHELS IN STØRE.

LAND IS TRADING AT 22 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 0

HØW MANY ACRES DØ YØU WISH TØ SELL? 0

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2000

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1000
BUT YØU HAVE ØNLY 100 PEØPLE TØ TEND THE FIELDS. NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1000
BUT YØU HAVE ØNLY 100 PEØPLE TØ TEND THE FIELDS. NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 999


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 2 , 0 PEØPLE STARVED, 13 CAME TØ THE CITY.
PØPULATIØN IS NØW 113
THE CITY NØW ØWNS 1000 ACRES.
YØU HARVESTED 5 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 5296 BUSHELS IN STØRE.

LAND IS TRADING AT 23 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 10

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2500

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1100
HAMURABI:  THINK AGAIN. YØU ØWN ØNLY 1010 ACRES.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1010


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 3 , 0 PEØPLE STARVED, 9 CAME TØ THE CITY.
PØPULATIØN IS NØW 122
THE CITY NØW ØWNS 1010 ACRES.
YØU HARVESTED 2 BUSHELS PER ACRE.
RATS ATE 515 BUSHELS.
YØU NØW HAVE 3566 BUSHELS IN STØRE.

LAND IS TRADING AT 26 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 20

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2750

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1030
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1020
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1000
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 9000\0\Ø
ILLEGAL NUMBER AT LINE 441
? 900
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 800
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 700
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 600
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 296 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 550


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 4 , 0 PEØPLE STARVED, 6 CAME TØ THE CITY.
PØPULATIØN IS NØW 128
THE CITY NØW ØWNS 1030 ACRES.
YØU HARVESTED 5 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 2771 BUSHELS IN STØRE.

LAND IS TRADING AT 26 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 0
HØW MANY ACRES DØ YØU WISH TØ SELL? 230

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2700
```

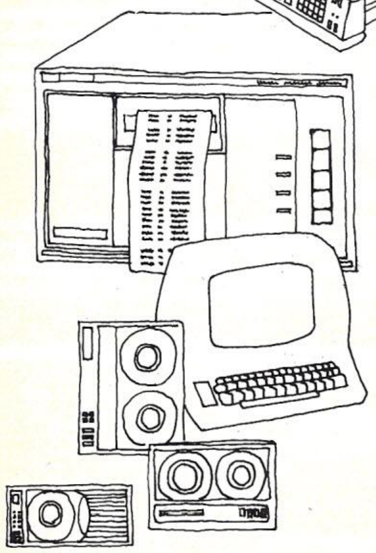
```
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED?" 900
HAMURABI:  THINK AGAIN. YØU ØWN ØNLY 800 ACRES.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 800


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 5 , 0 PEØPLE STARVED, 2 CAME TØ THE CITY.
PØPULATIØN IS NØW 130
THE CITY NØW ØWNS 800 ACRES.
YØU HARVESTED 4 BUSHELS PER ACRE.
RATS ATE 2825 BUSHELS.
YØU NØW HAVE 6026 BUSHELS IN STØRE.

LAND IS TRADING AT 19 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 100

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2750

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 900


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 6 , 0 PEØPLE STARVED, 4 CAME TØ THE CITY.
PØPULATIØN IS NØW 134
THE CITY NØW ØWNS 900 ACRES.
YØU HARVESTED 3 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 3626 BUSHELS IN STØRE.

LAND IS TRADING AT 18 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 100

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2750
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 1826 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 1500

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1000
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 326 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 700
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 326 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 652
HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 326 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 640


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 7 , 59 PEØPLE STARVED, 8 CAME TØ THE CITY.
PØPULATIØN IS NØW 83
THE CITY NØW ØWNS 1000 ACRES.
YØU HARVESTED 2 BUSHELS PER ACRE.
RATS ATE 3 BUSHELS.
YØU NØW HAVE 1283 BUSHELS IN STØRE.

LAND IS TRADING AT 23 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 0
HØW MANY ACRES DØ YØU WISH TØ SELL? 250

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 300000000
ILLEGAL NUMBER AT LINE 411
? 1600

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 750


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 8 , 3 PEØPLE STARVED, 11 CAME TØ THE CITY.
PØPULATIØN IS NØW 91
THE CITY NØW ØWNS 750 ACRES.
YØU HARVESTED 3 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 7308 BUSHELS IN STØRE.

LAND IS TRADING AT 23 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 0
HØW MANY ACRES DØ YØU WISH TØ SELL? 0

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 1875

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 750


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 9 , 0 PEØPLE STARVED, 3 CAME TØ THE CITY.
PØPULATIØN IS NØW 94
THE CITY NØW ØWNS 750 ACRES.
YØU HARVESTED 1 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 5808 BUSHELS IN STØRE.

LAND IS TRADING AT 18 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 100

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 1880

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 850


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 10 , 0 PEØPLE STARVED, 8 CAME TØ THE CITY.
PØPULATIØN IS NØW 102
THE CITY NØW ØWNS 850 ACRES.
YØU HARVESTED 5 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 5953 BUSHELS IN STØRE.

LAND IS TRADING AT 25 BUSHELS PER ACRE.
HØW MANY ACRES DØ YØU WISH TØ BUY? 150

HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE? 2040

HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 1000


HAMURABI:  THINK AGAIN. YØU HAVE ØNLY
 163 BUSHELS ØF GRAIN.  NØW THEN,
HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED? 321


HAMURABI:  I BEG TØ REPØRT TØ YØU,
IN YEAR 11 , 0 PEØPLE STARVED, 8 CAME TØ THE CITY.
PØPULATIØN IS NØW 110
THE CITY NØW ØWNS 1000 ACRES.
YØU HARVESTED 1 BUSHELS PER ACRE.
RATS ATE 0 BUSHELS.
YØU NØW HAVE 324 BUSHELS IN STØRE.

IN YØUR 10-YEAR TERM ØF ØFFICE, 5.44506 PERCENT ØF THE
PØPULATIØN STARVED PER YEAR ØN AVERAGE, I.E., A TØTAL ØF
 62 PEØPLE DIED!!
YØU STARTED WITH 10 ACRES PER PERSØN AND ENDED WITH
 9.09091 ACRES PER PERSØN.

YØUR PERFØRMANCE CØULD HAVE BEEN SØMEWHAT BETTER, BUT
REALLY WASN'T TØØ BAD AT ALL.  20 PEØPLE WØULD
DEARLY LIKE TØ SEE YØU ASSASSINATED BUT WE ALL HAVE ØUR
TRIVIAL PRØBLEMS.

SØ LØNG FØR NØW.


READY
```



(Continued on p. 82)

PROGRAM

```
HMRABI  07:38 PM          14-SEP-76
10 REM *** CONVERTED FRØM THE ØRIGINAL FØCAL PRØGRAM AND MØDIFIED FØR
20 REM *** EDUSYSTEM 70 BY DAVID AHL, DIGITAL
80 PRINT "TRY YØUR HAND AT GØVERNING ANCIENT SUMERIA"
85 PRINT "SUCCESSFULLY FØR A 10-YR TERM ØF ØFFICE.":PRINT
90 RANDØMIZE:LET D1=0:LET P1=0
100 LET Z=0:LET P=95:LET S=2800:LET H=3000:LET E=H-S
110 LET Y=3:LET A=H/Y:LET I=5:LET Q=1
210 LET D=0
215 PRINT:PRINT:PRINT "HAMURABI:  I BEG TØ REPØRT TØ YØU,":LET Z=Z+1
217 PRINT "IN YEAR"Z","D"PEØPLE STARVED,"I"CAME TØ THE CITY."
218 LET   P=P+I
227 IF Q>0 THEN 230
228 LET P=INT(P/2)
229 PRINT "A HØRRIBLE PLAGUE STRUCK!  HALF THE PEØPLE DIED."
230 PRINT "PØPULATIØN IS NØW"P
232 PRINT "THE CITY NØW ØWNS"A"ACRES."
235 PRINT "YØU HARVESTED"Y"BUSHELS PER ACRE."
250 PRINT "RATS ATE"E"BUSHELS."
260 PRINT "YØU NØW HAVE"S"BUSHELS IN STØRE.":PRINT
270 IF Z=11 THEN 860
310 LET C=INT(10*RND(0)):LET Y=C+17
312 PRINT "LAND IS TRADING AT"Y"BUSHELS PER ACRE."
320 PRINT "HØW MANY ACRES DØ YØU WISH TØ BUY";
321 INPUT Q:IF Q<0 THEN 850
322 IF   Y*Q<=S THEN 330
323 GØSUB 710
324 GØTØ 320
330 IF Q=0 THEN 340
331 LET A=A+Q:LET S=S-Y*Q:LET C=0
334 GØTØ 400
340 PRINT "HØW MANY ACRES DØ YØU WISH TØ SELL";
341 INPUT Q:IF Q<0 THEN 850
342 IF Q<A THEN 350
343 GØSUB 720
344 GØTØ 340
350 LET A=A-Q:LET S=S+Y*Q:LET C=0
400 PRINT
410 PRINT "HØW MANY BUSHELS DØ YØU WISH TØ FEED YØUR PEØPLE";
411 INPUT Q
412 IF Q<0 THEN 850
418 REM *** TRYING TØ USE MØRE GRAIN THAN IN THE SILØS?
420 IF Q<=S THEN 430
421 GØSUB 710
422 GØTØ 410
430 LET S=  S-Q:LET C=1:PRINT
440 PRINT "HØW MANY ACRES DØ YØU WISH TØ PLANT WITH SEED";
441 INPUT D:IF D=0 THEN 511
442 IF D<0 THEN 850
444 REM *** TRYING TØ PLANT MØRE ACRES THAN YØU ØWN?
445 IF D<=A THEN 450
446 GØSUB 720
447 GØTØ 440
449 REM *** ENØUGH GRAIN FØR SEED?
450 IF INT(D/2)<S THEN 455
452 GØSUB 710
453 GØTØ 440
454 REM *** ENØUGH PEØPLE TØ TEND THE CRØPS?
455 IF D<10*P THEN 510
460 PRINT "BUT YØU HAVE ØNLY"P"PEØPLE TØ TEND THE FIELDS. NØW THEN,"
470 GØTØ 440
510 LET S=S-INT(D/2)
511 GØ  SUB 800
512 REM *** A BØUNTYFULL HARVEST!!
515 LET Y=C:LET H=D*Y:LET E=0
521 GØSUB 800
522 IF INT(C/2)<>C/2 THEN 530
523 REM *** THE RATS ARE RUNNING WILD!!
525 LET E=INT(S/C)
530 LET S=S-E+H
531 GØSUB 800
532 REM *** LET'S HAVE SØME BABIES
533 LET I=INT(C*(20*A+S)/P/100+1)
539 REM *** HØW MANY PEØPLE HAD FULL TUMMIES?
540 LET C=INT(Q/20)
541 REM *** HØRRØRS, A 15% CHANCE ØF PLAGUE
542 LET Q=INT(10*(2*RND(0)-.3))
550 IF P<C THEN 210
551 REM *** STARVE ENØUGH FØR IMPEACHMENT?
552 LET D=P-C  :IF D>.45*P THEN 560
553 LET P1=((Z-1)*P1+D*100/P)/Z
555 LET P=C:LET D1=D1+D:GØTØ 215
560 PRINT:PRINT "YØU STARVED"D"PEØPLE IN ØNE YEAR!!!"
565 PRINT "DUE TØ THIS EXTREME MISMANAGEMENT YØU HAVE NØT ØNLY"
566 PRINT "BEEN IMPEACHED AND THRØWN ØUT ØF ØFFICE BUT YØU HAVE"
567 PRINT "ALSØ BEEN DECLARED 'NATIØNAL FINK' !!":GØTØ 990
710 PRINT "HAMURABI:  THINK AGAIN. YØU HAVE ØNLY"
711 PRINT S"BUSHELS ØF GRAIN.  NØW THEN,"
712 RETURN
720 PRINT "HAMURABI:  THINK AGAIN. YØU ØWN ØNLY"A"ACRES.  NØW THEN,"
730 RETURN
800 LET C=INT(RND(0)*5)+1
801 RETURN
850 PRINT:PRINT "HAMURABI:  I CANNØT DØ WHAT YØU WISH."
855 PRINT "GET YØURSELF ANØTHER STEWARD!!!!!"
857 GØTØ 990
860 PRINT "IN YØUR 10-YEAR TERM ØF ØFFICE,"P1"PERCENT ØF THE"
862 PRINT "PØPULATIØN STARVED PER YEAR ØN AVERAGE, I.E., A TØTAL ØF"
865 PRINT D1"PEØPLE DIED!!":LET L=A/P
870 PRINT "YØU STARTED WITH 10 ACRES PER PERSØN AND ENDED WITH"
875 PRINT L"ACRES PER PERSØN.":PRINT
880 IF P1>33 THEN 565
885 IF L<7 THEN 565
890 IF P1>10 THEN 9  40
892 IF L<9 THEN 940
895 IF P1>3 THEN 960
896 IF L<10 THEN 960
900 PRINT "A FANTASTIC PERFØRMANCE!!!  CHARLEMANGE, DISRAELI, AND"
905 PRINT "JEFFERSØN CØMBINED CØULD NØT HAVE DØNE BETTER!":GØTØ 990
940 PRINT "YØUR HEAVY-HANDED PERFØRMANCE SMACKS ØF NERØ AND IVAN IV."
945 PRINT "THE PEØPLE (REMAINING) FIND YØU AN UNPLEASANT RULER, AND,"
950 PRINT "FRANKLY, HATE YØUR GUTS!":GØTØ 990
960 PRINT "YØUR PERFØRMANCE CØULD HAVE BEEN SØMEWHAT BETTER, BUT"
965 PRINT "REALLY WASN'T TØØ BAD AT ALL. "INT(P*  .8*RND)"PEØPLE WØULD"
970 PRINT "DEARLY LIKE TØ SEE YØU ASSASSINATED BUT WE ALL HAVE ØUR"
975 PRINT "TRIVIAL PRØBLEMS."
990 PRINT:FØR N=1 TØ 10:PRINT CHR$(7);:NEXT N
995 PRINT "SØ LØNG FØR NØW.":PRINT
999 END
```



Mass ROM for $500?

A holographic Read-Only-Memory system from Holofile Industries Ltd. in Los Angeles may have a major impact on personal computing within the next couple of years.

Its device, the Holofile Memory, can read 200 million bits of data from a single microfiche. If this L.A. phonebook-sized capacity is inadequate, the user can slip in as many more fiches as he needs.

The reader offers one-second access (worst case) to any of several hundred records on the fiche, then extracts data at a rate up to five megabits per second. The memory can interface readily to microcomputers or to other peripherals. A smart terminal might be able to address this large file directly.

The exciting feature of the Holofile system is its probable cost — $500 or less, retail, for the reader. That's within the personal computing range.

Recording is another matter, of course. The recorders are likely to cost "several hundred thousand dollars," according to the company's literature (though Jim Case of the company commented in a telephone interview that the company is expecting $1.5 million of capital invested in production to provide them with "two or three recorders and perhaps 5000 readers" ready for sale). Chances are that most recording will be handled by service centers that transfer customer data to Holofile form for a fee.

Once the customer has a master fiche containing his data, he can have that master reproduced in quantity for 10 to 15 cents.

We can readily imagine distribution of a "magazine" in this form, containing articles, documentation and directly usable programs and data.

It's a chicken-and-egg problem,

The Holofile Memory. 200 million bits of data on a single microfiche.

of course. Publication of material for wide distribution isn't worthwhile until enough people are equipped to receive it, and it's not worthwhile to become equipped unless there's something to receive. However, the systems may prove practical within big organizations fairly soon and they might spill enough recorded material (deliberately as well as accidentally) to stimulate the personal computer market.

Holofile reached into giant TRW's closetful of developed, but shelved, technology to come up with this commercial application of exotic technology. If it pays off for both companies, they and others may be encouraged to rummage through the closets for other systems whose applications will enhance personal computing.

The price looks right. Performance in the market has yet to be measured. Keep watching.

Computer for $5

You can buy a complete computer system — including CPU, memory and I/O peripherals — for $5. You can assemble the kit in five minutes — it requires no soldering. You can understand the instruction manual even if you've never played with electronics or computers before.

Developed at Bell Telephone Laboratories, the system uses strange new hardware. Each IC connects to other devices **without wiring.**

The random-access memory is pencil-erasable. To access it, the system employs high-speed Optical Character Recognition.

The elaborate console displays the contents of each register and the entire contents of main memory simultaneously. The Program Counter, in the shape of a ladybug, hops around the console, to show all-too-graphically where it is at each moment.

The CPU boasts hardware decimal arithmetic to eliminate time-consuming binary-to-decimal conversions. Each "cell" (memory location) holds three decimal digits and has a two-digit address. The computer uses no bits or bytes.

The price includes even a card reader and lineprinter.

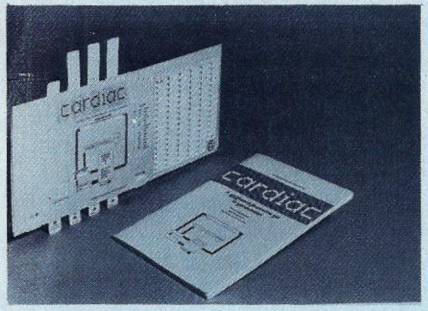
It is called the Cardiac computer. I own one myself, and it works quite well.

To bring the price down to $5, Bell Labs took one short-cut: the entire computer is made out of cardboard. To assemble it, use Scotch tape or glue. The "IC" stands for Integrated Cardboard. The pencil-erasable memory is created and erased with your own pencil and relies on your eyes (the "very-high-speed Optical Character Recognition"). The card read-

random access

er and lineprinter use you as the power supply. So does the CPU. The CARDIAC is a Cardboard Illustrative Aid to Computation.

Its 10 op codes are pleasantly simple. By playing with them,



$5 Cardiac computer

you'll learn machine- and assembly-language programming faster than on a more traditional machine. Here are the op codes:

OXY	If the card reader is empty, move the ladybug to cell 00 and stop. Otherwise, copy from the card to cell XY and advance the card.
1XY	Copy cell XY's contents into the accumulator.
2XY	Add cell XY's contents into the accumulator.
3XY	If the number in the accumulator is positive, move the ladybug to cell XY.
4XY	Shift the number in the accumulator to the left, X places, then to the right, Y places.
5XY	Copy from cell XY to the lineprinter. Then advance the lineprinter's paper.
6XY	Copy the accumulator's contents into cell XY.
7XY	Subtract cell XY's contents from the accumulator.
8XY	Copy the ladybug's location into cell 99 (after the 8 which is in cell 99 permanently). Then move the ladybug to cell XY.
9XY	Move the ladybug to cell XY and stop.

To counteract a slight "hardware defect", cross out the sign "Is Input Card Blank?" and rewrite it above the words "Yes" and "No". Mail $4.95 to Cardiac's distributor: Comspace Corp., 350 Great Neck Rd., Farmingdale, N.Y. 11735. —RW

Small fry

National Semiconductor is doing exciting things with SC/MP (yes, they call it Scamp), its little 8-bit computer on a card that sells for $99 in kit form, $125 assembled. Until now, Scamp has needed a Teletype for I/O, but National is offering a nifty new little $95 keyboard kit that gives the operator a hex keyboard, command keys and a six-digit hex display in a package that looks like a pocket calculator. It's connected to its circuit-cardful of buffers, decoder driver multiplexer, etcetera, by a 21-wire ribbon cable.

National says, "The heart of the Keyboard Kit is a ROM firmware package (512 bytes), which allows the hexadecimal keyboard to execute programs, examine or modify the contents of memory and the SC/MP registers and monitor program performance." Scamp is growing more flexible as the cost of the system diminishes.

A rumor predicts that National will shortly announce Scamp BASIC in 4K of ROM, using 2K of RAM.

Please your Kim

Use your Kim-1 to play Mastermind, puzzle you about Shooting Stars, measure your reaction time, test whether you're drunk, turn into an old-fashioned clock (hours, minutes, seconds) or adding machine and do other neat tricks. How? Send $10 to MicroCosmos to get software on a cassette, source listings, operating instructions and hints on writing your own programs. All programs run on a basic Kim-1: you don't need additional memory or a Teletype.

The package is called Please. Bob Tripp, who heads the new company, explains: "The name does not have any special meaning except that we hope it will please its users, that it is nice to be polite even to your computer and, finally, think of all the free advertising we get. People are always saying 'Please.' "

MicroCosmos is at 210 Daniel Webster Highway South, S. Nashua, N.H. 03060.

Sobering up on high

While many enthusiasts are confident that the personal computing revolution is already beginning to sweep the world, they may be sobered to discover how big the world is and how conspicuous computers are by their absence from logical places.

Example: study the gift catalogs offered by the major airlines in the seat pockets of the planes. In one recent catalog a great array of electronic items is offered, ranging from simple, practical products like radios and small pocket calculators to gold-plated luxury items whose practicality is secondary to their richness.

Cameras, television sets, hi-fi systems and CB transceivers are for sale at prices from $15 to $260. At the top of the luxury list, at five cents less than $800, is a calculator watch with an eight-digit LED display.

You can buy a lot of computer for $800. It's a prestige gift item, certainly. The introduction of microcomputers to such a catalog would be an event of some real note. Microcomputers haven't been noticed.

No occasion for gloom that our sophisticated technical revolution hasn't even swept the rich, sophisticated airline market yet — considering all the excitement already, the coming action should be satisfyingly dramatic.

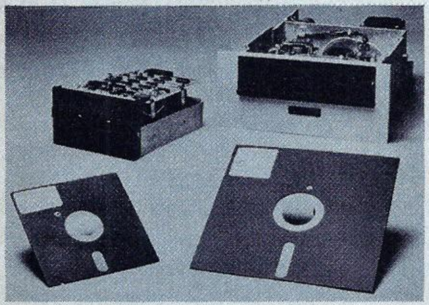
It's sobering, all the same.

Minifloppies replace floppy disks for lower cost random access

If you can't afford a floppy disk drive, try Shugart Associates' Minifloppy disk drive. It's cheaper, more compact and almost as good.

While floppy drives range from $625 to $3700 (depending on the manufacturer and model), the Minifloppy drive costs only $390; in oem quantities it costs $250. While floppy drives range from 12 pounds to 90, the Minifloppy drive weighs only 3 pounds. Its size (3.25" high, 5.75" wide and 8" long) and power requirements (15 watts continuous duty, 7.5 watts standby) also undercut all floppy drives.

The Minifloppy's disks are smaller. While standard floppy disks are 7.88" in diameter, the Minidiskette is 5.125". While standard floppy disks hold 242 kilobytes if soft-sectored, 315 if hard-sectored and even more if two-sided or



Minifloppy drive and diskette are on the left, standard units on the right.

double-density, the Minidiskette holds 80. (The kilobyte figures are for formatting into 128-byte sectors; you can store a few more kilobytes if you're willing to put up with 256-byte sectors.) While standard one-sided single-density disks cost between $7 and $7.50 (in quantities of 10), a Minidiskette costs $4.50 in quantities of 10 and $3.50 in oem quantities. Ferrell Sanders, the firm's marketing director, predicts small diskettes will eventually sell for $3.

The Minifloppy drive is slower and less accurate than standard floppy drives. Its average access time is 566 milliseconds instead of 350. It transfers 12 kilobytes per second instead of 31. Its (soft) error rate is 1 bit per 10^8 instead of 1 bit per 10^9.

Its long average access time is caused by slow rotation (300 rpm instead of 360) and slow arm movement (1.36 seconds to move from the outermost track to the innermost instead of 0.76).

Though not as fast and accurate as standard floppy drives, it is nevertheless satisfactory for a personal computer, and its low price will make it popular.

If your're new at this game, we should warn you that the price of a drive (for a floppy or Minifloppy or tape or whatever) does **not** include the controller you'll need for interfacing.

Shugart Associates is at 435 Indio Way, Sunnyvale, Calif. 94086.

$599 minifloppy package for Altair, Imsai, Z-80

includes drive, interfacing, floppies and new BASIC

Shugart's Minifloppy drive is useless without a controller and software. North Star Computers Inc. will sell you a Minifloppy drive, controller, interfacing and two Minidiskettes (one of which contains a Disk Operating System and Minifloppy BASIC) for $599 in kit form, $699 assembled. To use North Star's package, you must own an Altair 8800, Imsai 8080 or some other computer using an 8080 or Z-80 CPU.

Minifloppy BASIC is more powerful than MITS 8K BASIC, though not as powerful as MITS 12K. The IF statements can contain AND, OR, NOT, ELSE, arbitrary BASIC statements, and be nested. Built-in string functions are provided, or you can create your own, using multiple-line definitions. The PRINT USING statement permits FORTRAN-like editing: 5F2 makes the computer print a number having 5 characters, 2 of them after the decimal point. You can output to any port (by saying OUT) and pass arguments to assembly-language subroutines.

Minifloppy BASIC handles sequential and random-access files. Even in a random-access file, a record can contain a mixture of numbers and strings.

The punctuation differs from MITS'. The word PRINT is abbreviated by an exclamation point instead of a question mark. Statements on the same line are separated by a back-slash instead of a colon. But you can still type MITS' punctuation; the computer will automatically convert it to the new punctuation, which you'll see on the listing.

The standard version is accurate to eight significant digits. For a custom version, send North Star any other even number and $25. But remember that higher accuracy requires more of the computer's time and memory, and functions such as SIN and LOG are limited to 14 significant digits.

Although numeric variables can be subscribed, string variables must not. You must not say MAT.

The controller is a printed circuit board, whose PROM includes bootstrap software that makes your whole machine operative as soon as you turn the power on. The controller can handle up to three drives; the second and third

drives cost $425 each, including cables.

The package does not include a power supply, because power requirements (0.5 amp at +5 volts and 0.9 amp at +12 volts) can usually be fulfilled by the power supply that came with your Altair or Imsai. But North Star will sell you a power supply and cabinet as extras if you wish.

North Star formats the Minidiskette into 256-byte sectors, so you can store 89.6 kilobytes. Although Shugart sells Minidiskettes only in boxes of 10 ($45), North Star will sell them individually, for $4.50 apiece, as a favor to people who bought the North Star package.

The kits are expected to be available December 15, in time for Christmas. Assembled units will be available in January.

The assembled units have a 6-month warranty that covers everything except abuse. Kits have a 3-month warranty on all parts. If you put together your kit somewhat incorrectly, North Star will charge at most $30 to fix it, unless your error damaged hardware.

North Star, at 2465 4th St., Berkeley, Calif. 94710, accepts cash payments, BankAmericard charges or C.O.D. (with a 25% deposit and C.O.D. charges). Shipping is free. —RW

The way in San Jose

If you were in charge of helping a big school system prepare for the time when every high-school graduate must be competent to deal with computers, what would you do with a hard-won, little budget of $25,000 for equipment?

Dr. Peter Grimes, of the San Jose, Calif. Unified School District reports that his office is buying 10 Imsai computers and one Polymorphic Systems machine, with terminals for all. The machines are to be used with a class of "gifted" kids in elementary school. Grimes sees a time when all instruction about computing puts one kid at one terminal with one computer handling at least BASIC in 8K.

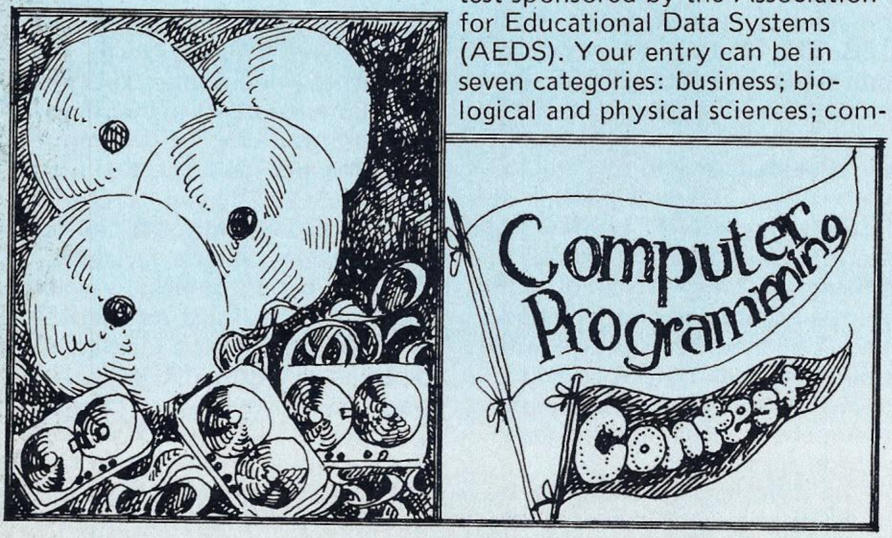
Why start with the gifted? No special bias, says Grimes, but that's where the money is now.

San Jose is pressing forward with a training program in which 30 nervous teachers will assemble computer kits, learn to use them and put the assembled systems out in the schools for the kids. The chosen small system is the Data Handler, based on the 6502 chip. Grimes likes it, because it works with Teletype I/O, has a BASIC language and is inexpensive — about $160 plus a $40 power supply.

"San Jose is trying to prepare for the Deluge," says Grimes. "My role is to get started in any way possible, and they're giving me wholehearted support."

For years San Jose had offered students timesharing terminals, but "there's no point in teaching programming without full access to computers. New technology and dropping cost are giving us a chance now to reach a lot of students." Grimes grows enthusiastic: "By 1984 computers will be under personal control, not run by Big Brother exclusively."

We'll be hearing more from San Jose.



For students only

If you're a student in grades 7-12, consider entering the 14th Annual Computer Programming Contest sponsored by the Association for Educational Data Systems (AEDS). Your entry can be in seven categories: business; biological and physical sciences; computer art; computer science; games; humanities; and mathematics. The Grand Prize Winner gets a $100 Savings Bond plus a free trip to the 1977 AEDS Convention in Fort Worth, Tex. on April 25-29. The winner's sponsor can go too, all expenses paid.

Each category winner gets a $50 Savings Bond. Many of the participants will receive one-year subscriptions to professional publications.

You can work in a group. But if your group wins, only one of you goes to Fort Worth.

For details, write to AEDS Programming Contest, Dr. Jane Donnelly Gawronski, Dept. of Education, San Diego County, 6401 Linda Vista Road, San Diego, Calif. 92111.

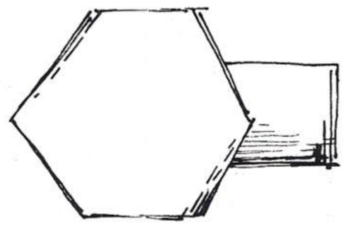
With Colonel Colt's six-gun, the uneasy average man was equal to the dangers of the frontier. The uneasy average man at today's frontiers of knowledge can hold his own with the personal computer. It is:

THE EQUALIZER

In 1964 when I was walking down a corridor toward the office of a publisher from whom I hoped doubtfully to collect an overdue check, a hand reached out of the wall and dragged me into a meeting.

Three editors and an artist were wrangling over the selection of symbols for the covers of books in an educational series treating history, mathematics literature and biology. The artist's work was unsatisfactory, but the editors couldn't quite explain why. As an innocent passerby, unbiased by any knowledge of their argument, I was nominated to give an objective opinion. To make me feel perfectly at ease while I dawdled uncertainly over the pictures, they all glared at me in hostile silence.

Three of the symbols the artist had rendered were plain enough. The fourth baffled me. It looked something like this:



When I asked what it was, the outraged artist barked: "It's a virus, a stylized virus. It symbolizes basic biology."

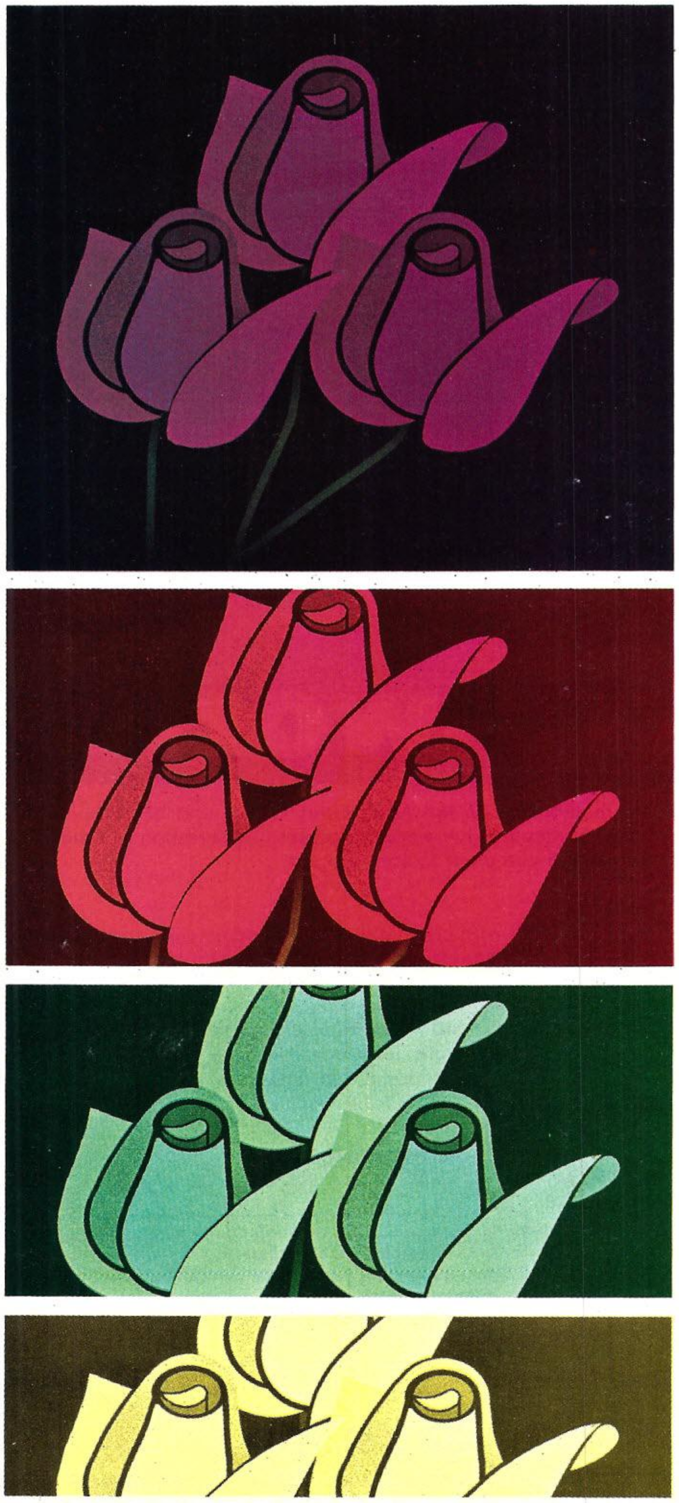
"Oh," I said, embarrassed. "It looks so geometric, I thought it might have something to do with math."

"That's it!" said an editor. "My idea of a symbol for biology is something that looks biological, like a deer eating a rose."

The artist slammed the material to the floor, and a great hubbub of argument burst out as I sneaked away toward the office again. The artist must have jumped out the window, raced me there and hidden my check, because nobody could find it.

I doubt that computers will ever resolve publishers' late payment problems. But computers probably will help with their art problems. How handy it would be if an editor who can't draw and is largely inarticulate in expressing his wishes for effective illustrations could sit down at a computer console and get the machine's help in sketching ideas. Even if these computer-generated pictures aren't "finished art," they may let the desperate editor convey a sense of his taste and meaning to the eager but frustrated artist.

The artist would also like a flexible tool that helps him render his ideas quickly. Of the two standard approaches in the computer systems, the first is to feed the computer a picture that it breaks into bits for storage in a memory from which the picture can be recalled. The artist may then ask the computer to fetch out a picture and display it on a television screen (possibly in good-quality high resolution) in any of various sizes and colors. He may ask for distortions, altered angles and perspectives, combinations of different figures, backgrounds, lettering, textures and fancies of all sorts. This approach calls for a lot of computer, display gear and memory, but it's possible.



Genigraphics by Glenn Brackett

See article describing the Genigraphics system on page 41.

The second approach lets the artist start from scratch on a plain background and use the computer as a controllable automatic pencil to draw rapidly, make alterations quickly and store the image for recall. This technique takes not quite so much computer, depending on what's desired.

There must be some clever, inexpensive, alternative approaches not yet conceived that will lift the curse and answer our prayers for something practical.

Unfortunately, development of graphics systems has been almost entirely in the hands of people who understand something about computers but nothing about commercial art. The artists are stuck with whatever the computerfolk give them. So far, the pickings for artists have been mighty lean.

A dozen years ago, Paul Honore and I offered "computer art" prints for sale at San Francisco International Airport. (Were we the first with "digital art?" Probably not, but we were early.) We obtained materials from computer folk, printed the pictures at hideous expense and sent them to the airport newsstands on consignment. We lost only a few hundred dollars, and that's better than most early adventures with computer work for the common man.

The pictures were fancy geometric forms, of course, with zillions of fine lines that no human artist except M.C. Escher would fool with but which the computer loved to draw. All sorts of geometric forms, orderly, random, colored, negative, dense, long, wide. The first few hundred patterns were interesting. The next few thousand were not.

In the mid-1960's we were treated to lovely motion pictures John Whitney animated with a computer. His pulsating geometric forms, set to fascinating music, were delightful and exciting . . . the first dozen times through. Even the animated geometric humanoid figures produced by Cornell Aeronautical Labs and Boeing, among others, moved with a stylish grace that pleased the viewer only briefly, though engineers may have found a lingering utilitarian interest in the material after the rest of us had fallen asleep.

We all looked forward 10 years to times when artists and editors would have personal control of the computers and could make the world bright with useful and novel images that people would enjoy as they enjoy conventional artwork.

And what was the exciting graphics demonstration 10 years later at the 1976 World Altair Computer Conference? One of John Whitney's lovely 10-year-old movies, that's what. There has been little discernable progress in the field in a decade.

Whitney is still stuck with those everlasting geometric figures that the computer loves. John is game, affable and inventive. He explains doggedly that images, like music, have a fundamental mathematical relationship with human perception, but . . . sigh . . . ho-hum . . . zzzzzzzzzzz.

Computer art still mostly takes a form that pleases the dumb, tasteless computer, while the market for material that pleases dumb, tasteless editors grows constantly.

Personal computing may break the impasse if only some *artists* get their hands on computer systems and teach them to draw real pictures of real things that people recognize and like. Obviously, the matter can't be left to the computer experts with any hope of progress.

Personal computing gives us renewed hope of producing computer art for people instead of computer art for computers. Forget crash educational programs to teach us all to appreciate abstract, stylized images of who-knows-what. How about asking the computer to draw pictures of a deer eating a rose? There's this editor who . . .

—NBW III

CLUBS

ALABAMA

North Alabama Computer Club
c/o Jack Crenshaw
1409 Blevins Gap Road S.E.
Huntsville, AL 35802

CALIFORNIA

Bay Area Microprocessor Users Group
4565 Black Avenue
Pleasanton, CA 94566

California Computer Society
Sheila Clark
Box 9459
North Hollywood, CA 91609

Computer Org. of L.A. (COLA)
Box 43677
Los Angeles, CA 90043

Computer Phreaques Unit
c/o Mac McCormick
2090 Cross Street
Seaside, CA 93955

Douglas L. Penrod
1445 La Cima Road
Santa Barbara, CA 93101

Glendale Community College Computer Club
c/o V.S. Lashleu
1500 N. Verdugo Road
Glendale, CA 92108

Homebrew Computer Club
Box 626
Mountain View, CA 94040

Jim McCord
330 Vereda Legenda
Goleta, CA 93017

John T. Craig
2497 Lompoc-Casmalia Road
Lompoc, CA 93436
(People owning Varian Computers)

Litton Calculator/Computer Club
Litton Guidance and Control Systems
MS 78/31
5500 Canoga Avenue
Woodland Hills, CA 91364

LLLRA Hobbyist Group
c/o Charles D. Hoover
35 West Essen Street
Stockton, CA 95204

LO*OP Center
8099 La Plaza
Cotati, CA 94928

Sacramento Minicomputer Users Group
Box 741
Citrus Heights, CA 95610

San Diego Computing Society
Box 9988
San Diego, CA 92109

San Gabriel SCCS
c/o Dan Erickson
400 S. Cataline Avenue
Pasadena, CA 91106

Santa Barbara Computer Group
c/o Glenn A. McComb
210 Barrunca, Apt. 2
Santa Barbara, CA 93101

Santa Barbara Nameless Computer Club
c/o Doug Penrod
1445 La Clima Road
Santa Barbara, CA 93101

Southern California Computer Society
Box 987
South Pasadena, CA 91030

Southern California Computer Society
Box 3123
Los Angeles, CA 90051

29 Palms California Area Group
c/o Sgt. Wesley Isgrigg
74055 Casita Drive
29 Palms, CA 92277

UCLA Computer Club
3514 Boelter Hall
UCLA
Los Angeles, CA 90024

Valley Chapter, SCCS
c/o R. Stuart Gibbs
5652 Lemona Avenue
Van Nuys, CA 91411

Ventura County Computer Club
4409 Vinyard Avenue
Oxnard, CA 93030

COLORADO

Denver Amateur Computer Society
Box 6338
Denver, CO 80206

CONNECTICUT

Amateur Computer Society
Stephen B. Gray
260 Noroton Ave.
Darien, CT 06820

Connecticut Microists
c/o George Ahmuty
6011 Wendy Lane
Westport, CT 06881

So. New England Computer Society
c/o Charles Floto
267 Willow Street "1"
New Haven, CT 06511

DISTRICT OF COLUMBIA

Washington Amateur Computer Society
Robert Jones
4201 Massachusetts Avenue, Apt. 168W
Washington, DC 20016

FLORIDA

Miami Area Computer Club
c/o Terry Williamson
Box 430852, S.
Miami, FL 33143

Miami Computer Club
John Lynn
13431 S. W. 79th
Miami, FL 33183

Southern Florida Computer Group
c/o Roberto Denis
11080 N.W. 39th St.
Coral Springs, FL 33065

Tallahassee Amateur Computer Society
c/o Larry Hughes
Route 14, Box 351-116
Tallahassee, FL 32304

GEORGIA

Atlanta Microcomputer Hobbyist Club
c/o Jim Dunion
421 Ridgecrest Road
Atlanta, GA 30307

HAWAII

Aloha Computer Club
c/o Robert Kennedy
1541 Dominus No. 1404
Honolulu, HI 96822

ILLINOIS

Chicago Area Computer Hobbyists Exchange (CACHE)
Box 36
Vernon Hills, IL 60061

Chicago Microcomputer Users Group
c/o Bill Precht
1102 S. Edison
Lombard, IL 60148

Ice-Nine, Inc.
Box 291
Western Springs, IL 60558

INDIANA

Beta Iota Tau (Computer Fraternity)
c/o Richard R. Petke
R.H.I.T. Box 420
Terre Haute, IN 47803

Hoosier Amateur Computer and Kluge Society
c/o Ray Borill
111 S. College Avenue
Bloomington, IN 47401

Louisville Area Users of Microprocessors
115 Edgemont Drive
New Albany, IN 47150

KENTUCKY

Louisville Area Users of Microprocessors
c/o Steve Roberts (Cybertronics)
Box 18065
Louisville, KY 40218

LOUISIANA

New Orleans Computer Club
Amil Alline
1119 Pennsylvania Avenue
Slidell, LA 70458

MARYLAND

Chesapeake Microcomputer Club
236 St. David Court X4
Cockeysville, MD 21030

MASSACHUSETTS

Alcove Computer Club
c/o John P. Vullo
21 Sunset Avenue
North Reading, MA 01864

New England Computer Society
Box 198
Bedford, MA 01730

Richard F. Brown
730 Boston Post Road
Sudbury, MA 01776

MICHIGAN

Ann Arbor Computing Club
c/o Roger Gregory
1485 New Port Road
Ann Arbor, MI 48103

C. J. Lamesfield
Box 271
Davison, MI 48423

Computer Hobbyists Around Lansing
c/o Joyce and Marvin Church
4307 Mar Moor Drive
Lansing, MI 48917

Detroit Area Club
c/o Dennis Siemit
45466 Cluster
Utica, MI 48087

Detroit Area Users Group
c/o Dana Badertscher
18300 Ash
East Detroit, MI 48021

Mid-Michigan Micro Group
c/o William Serviss
13121 Tucker Drive
DeWitt, MI 48820

SEMCO
c/o Dick Weir
20,000 Great Oak Circle S.
Mt. Clemens, MI

MINNESOTA

Bit Users Association
Resource Access Center
3010 4th Avenue S.
Minneapolis, MN 55408

Southern Minnesota Amateur Computer Club
2212 N.W. 17th Avenue
Rochester, MN 55901

XXX-II Users Group
Dick Corner
514 S. 9th Street
Moorhead, MN 56560

MONTANA

Flathead Computer Society
Ernest G. Brooner
Box 236
Lakeside, MT 59922

NEW HAMPSHIRE

Nashua Area Computer Club
c/o Dwayne Jeffries
181 Cypress Lane
Nashua, NH 03060

New England Computer Club
70 Main Street
Peterborough, NH 03458

NEW JERSEY

Amateur Computer Group of N.J.
c/o Sol Libes
UCTI
Scotch Plains, NJ 07076

New Jersey Club
c/o Bruce C. Dalland
37 Brook Drive
Dover, NJ 07801

William Paulus
510 Nectar Avenue
Pine Hurst, NJ 08201

NEW MEXICO

Albuquerque Area Computer Club
Gary Tack Box 866
Corrales, NM 87048

NEW YORK

Buffalo Group
c/o Chuck Fischer
355 South Creek Drive
Depew, NY 14043

Ithaca Computer Club
c/o Steven Edelman
204 Dryden Road
Ithaca, NY 14850

Long Island Computer Club
c/o Popular Electronics
One Park Avenue
New York, NY 10016

New York Micro Hobbyist Group
c/o Robert Schwartz
375 Riverside Drive, Apt. 1E
New York, NY 10025

Pacesetter User's Group
1457 Broadway, Room 305
New York, NY 10016

Students Cybernetics Lab
16 Linwood Avenue
Buffalo, NY 14209

Westchester Amateur Computer Society
c/o Harold Shair
41 Colby Avenue
Rye, NY 10580

OHIO

Columbus Amateur Computer Society
c/o Walter Marvin
408 Thurber Drive West #6
Columbus, OH 43215

CLUBS

Cleveland Digital Group
c/o John Kabat, Jr.
1200 Seneca Blvd. #407
Broadway Heights, OH 44147

Dayton Computer Club
c/o Doug Andrew
8668 Sturbridge Avenue
Cincinnati, OH 45200

Midwest Alliance of Computer Clubs
c/o Gary Coleman
Box 83
Brecksville, OH 44141

OKLAHOMA

Central Oklahoma Amateur Computing Association (CENOACA)
c/o Lee Lilly
Box 2213
Norman, OK 73069

Oklahoma City Club
c/o Bill Cowden
2412 S. W. 45th
Oklahoma City, OK 73119

OREGON

Portland Computer Club
c/o Bill Marsh
2814 N.E. 40th Street
Portland, OR 97212

PENNSYLVANIA

Delaware Valley Chapter, SCCS
c/o Martin Dimmerman
1228 Barrowdale
Rydal, PA 19046

Pittsburgh Area Computer Club
c/o Fred Kitman OPUS-1
400 Smithfield Road
Pittsburgh, PA 15222

TEXAS

El Paso Computer Group
c/o Jack O. Coats, Jr.
213 Argonaut, Apt. 27
El Paso, TX 79912

Houton Amateur Microcomputer Club
c/o Al Corry, Librarian
Box 1644
Pasadena, TX 77501

NASA-JSC Computer Hobbyist Club
c/o Marlowe Cassetti
1011 Davenport
Seabrook, TX 77586

Texas A&M Univ. Microcomputer Club
Box M-9
Aggieland Station, TX 77844

Texas Computer Club
c/o L.G. Walker
Route 1, Box 272
Aledo, TX 76008

The Computer Hobbyist Group of North Texas
c/o Bill Fuller
2377 Dalworth 157
Grant Prairie, TX 75050

UTAH

Salt Lake City Computer Club
2925 Valley View Avenue
Holladay, UT 84117

VIRGINIA

Andrew Convery
2315 Freetown Court, Apt. 110
Reston, VA 22091

Dyna-Micro Users Group
c/o Dr. Frank Settle, Jr.
Digital Directions
Box 1053
Lexington, VA 24450

IBM 5100 Users Group
c/o Richard E. Easton, M.D.
5541 Parliment Drive, Suite 104
Virginia Beach, VA 23462

Peninsula Computer Hobbyist Club
c/o Larry Polis
2 Weber Lane
Hampton, VA 23663

Washington-Baltimore Computer Hobbyist Club
c/o Richard Rubinstein
7711 Elba Road
Alexandria, VA 22306

WASHINGTON

Northwest Computer Club
c/o Bob Wallace
Box 5304
Seattle, WA 98105

WISCONSIN

Wisconsin Area Tribe of Computer Hobbyists (WATCH)
c/o Don Stevens
Box 159
Sheboygan Falls, WI 53085

If you're moving...

Please allow at least six weeks for us to put your new address into our subscription computer. Send your new address and the mailing label with your old address to: Circulation Dept., Personal Computing, 167 Corey Rd., Brookline, MA 02146.

Or use this convenient form.

Please attach here the mailing label with your old address

Write in here your new address.

NAME ____________________

ADDRESS ____________________

CITY ________ STATE ________ ZIP ______

Personal Computing
Personal Computi
Personal Comp
Personal Con

Ad Index

Sales Offices

Northeast: Irwin Stern, 167 Corey Rd., Brookline, MA 02146; (617) 232-5470 • **Mid-Atlantic:** Robert Poggi, 69 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Mid-West:** Ralph Petersen, 1 Wheaton Ctr. No. 1706, Wheaton, IL 60187; (312) 653-2480 • **Southwest:** David Bunnell, 401 Louisiana, S.E. No. G, Albuquerque, NM 87108; (505) 266-1173 • **West:** Yuri Spiro & Carol Stagg, 823 Enchanted Way, Pacific Palisades, CA 90272; (213) 454-0624

See SwTPC computer equipment at your local dealer

CALIFORNIA

The Electric Brain Computer Store
700 Village Parkway, Suite L
Dublin, Ca. 94566
(415) 828-7480

Computerware
830 First St.
Encinitas, Ca. 92024
(714) 436-3512

Cyberdux
Microcomputer Applications
1210 Santa Fe Dr.
Encinitas, Ca. 92024
(714) 279-4189

A-VID Electronics Co.
1655 East 28th St.
Long Beach, Ca. 90806
(213) 426-5526

The Byte Shop Computer Store #1
1063 El Camino Real
Mountain View, Ca. 94040
(415) 969-5464

The Computer Center
8205 Ronson Rd.
San Diego, Ca. 92111
(714) 292-5302

Computer Store of San Francisco
1093 Mission St.
San Francisco, Ca. 94103
(415) 431-0640

The Byte Shop Computer Store #2
3400 El Camino Real
Santa Clara, Ca. 95051
(408) 249-4221

The Computer Store
820 Broadway
Santa Monica, Ca. 90401
(213) 451-0713

FLORIDA

Electronics for Yachting, Inc.
2001 S.W. 20th St.
Ft. Lauderdale, Fl. 33315
(305) 525-3479

Computer Assoc., Inc.
6900 N. Kendall Dr., Suite A103
Miami, Fl. 33156

Marsh Data Systems
5405-B Southern Comfort Blvd.
Tampa, Fl. 33614
(813) 886-9890

Microcomputer Systems, Inc.
144 S. Dale Mabry Ave.
Tampa, Fl. 33609
(813) 879-4301

GEORGIA

Atlanta Computer Mart
5091-B Buford Highway
Atlanta, Ga. 30340
(404) 455-0647

ILLINOIS

American Microprocessors
Equipment & Supply Corp.
241 Indian Creek Rd.
Prairie View, Il. 60069
(312) 634-0076

INDIANA

The Data Domain
111 S. College Ave.
Bloomington, In. 47401
(812) 334-3607

KANSAS

Midwest Scientific Instruments, Inc.
220 W. Cedar
Olathe, Ks. 66061
(913) 764-3273

KENTUCKY

Cybertronics
312 Production Ct.
Louisville, Ky. 40299
(502) 499-1551

LOUISIANA

Baxter's T.V.
7964 Jefferson Hwy.
Baton Rouge, La. 70809
(504) 924-5303

MARYLAND

The Computer Workshop, Inc.
5709 Fredrick Ave.
Rockville, Md. 20852
(302) 468-0455

MASSACHUSETTS

Computer Warehouse Store
584 Commonwealth Ave.
Boston, Ma. 02215
(617) 261-2701

Control Concepts
P.O. Box 272
Needham Heights, Ma. 02194

NEVADA

Johnson T.V., Inc.
2607 E. Charleston
Las Vegas, Nv. 89104
(702) 382-5553

NEW HAMPSHIRE

Aircom, Inc.
Route 16-B
Union, N.H. 03887
(603) 473-2323

NEW JERSEY

William Electronics Supply
1863 Woodbridge Ave.
Edison, N.J. 08817
(201) 985-3700

Midwest Enterprises, Inc.
815 Standish Ave.
Westfield, N.J. 07090
(201) 233-7090

NEW YORK

Synchro-Sound Enterprises
193-25 Jamica
Hollis, N.Y. 11423

Audio Design Electronics
487 Broadway, Room 512
New York, N.Y. 10013
(212) 226-2038

Computer Mart of New York, Inc.
314 Fifth Ave.
New York, N.Y. 10001
(212) 279-1048

Waco Trading Co., Inc.
239 Park Avenue South
New York, N.Y. 10003
(212) 674-0470

OHIO

ELS Systems
2209 N. Taylor Rd.
Cleveland Heights, Oh. 44112
(216) 249-7820

PENNSYLVANIA

Martin J. O'Boyle & Assoc.
P.O. Box 9094
Pittsburgh, Pa. 15224
(412) 361-1602

UTAH

The Computer Room
1455 South 1100 East
Salt Lake City, Ut. 84105
(802) 466-7911

TEXAS

Printing & Office Supply Co.
of Texas, Inc.
130 One Shell Plaza
Houston, Tx. 77002
(713) 228-8966

The Micro Store
634 S. Central Expressway
Richardson, Tx. 75080
(214) 231-4088

WASHINGTON

Retail Computer Store
410 N.E. 72nd St.
Seattle, Wa. 98115
(206) 524-4101

WISCONSIN

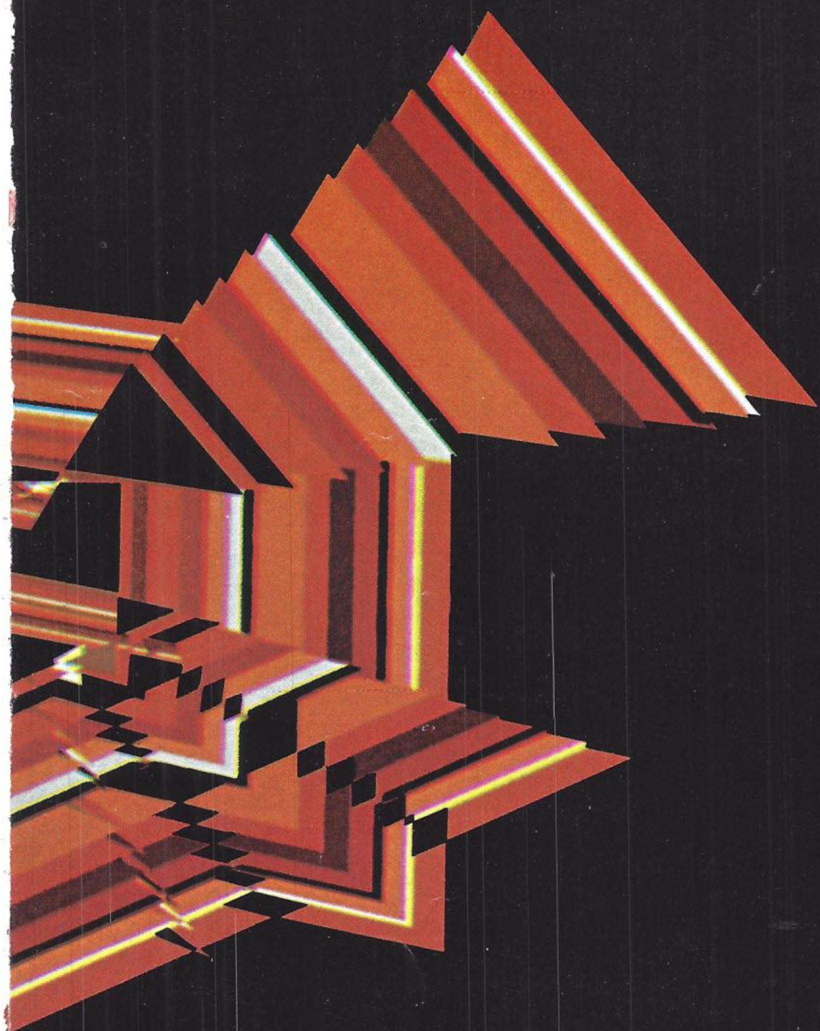
The Milwaukee Computer Store
6916 W. North Ave.
Milwaukee, Wi. 53213
(414) 259-9140

FOREIGN

Computer Resources
Chaussee de Charieroi, 80
1060 Brussels, Belgium

SDS Technical Devices, Ltd.
1138 Main St.
Winnipeg, Man. R2W 3F3 Canada
(204) 589-4803

The Entryphone Co., Ltd.
172 Ifield Rd.
London SW 10 9AG, England
01-373-8571



"*Abstract Orange*" by Kathleen McGlynn on a Genigraphic system.
Personal Computing Magazine January/February, 1977